all rights reversed

HACKER

Season 05 / Issue 02

τεύχος 050 - Sjokkerandi Edition

Φτάσαμε τα πενήντα [*]



Μετάδοση εικόνας μέσω ήχου

Το KVM για αληθινούς administrators

Φτιάξτε το δικό σας Netflix, με το Plex

Το πρόβλημα της εμπιστοσύνης στην κρυπτογραφία

Δωρεάν TLS certificates για το site σας, εδώ και τώρα

[*] ...κι ακόμα προθερμαινόμαστε ;)

all rights reversed

HACKER


CMYK & WEB
cmykandweb.gr
GRAPHIC DESIGN
design • layout • web

#dH050 | Περιεχόμενα

Εκδότης / Διευθυντής Χρήστος Βαρελάς

Αρχισυντάκτης Παναγιώτης Βαρελάς

Σύμβουλοι έκδοσης Νίκος Μουρατίδης, Βούλα Παυλίδου

Σύνταξη Παναγιώτης Βαρελάς, Χρήστος Βαρελάς, Γιώργος Γιάννου, Μανώλης Κιαγιάς, Πέτρος Κυλαδίτης, Νίκος Μουρατίδης, Θωμάς Σερμπίνης, Μέσος Παπαδόπουλος (TM), Χρίστος Τόμπρας, @arkoudos

Γραμματική επιμέλεια Βούλα Παυλίδου

Καλλιτεχνική Επιμέλεια Πέτρος Φιλιππίδης

Το ηλεκτρονικό περιοδικό deltaHacker είναι συνδρομητικό και εκδίδεται από την Parabing Creations δώδεκα φορές το χρόνο, κάθε μήνα. Η Parabing Creations έχει την έδρα της στην Καλαμαριά Θεσσαλονίκης. Ο δικτυακός τόπος του περιοδικού είναι στο http://deltahacker.gr. Το email επικοινωνίας είναι το talk2us@deltahacker.gr και ειδικά για τις συνδρομές είναι το subscriptions@deltahacker.gr. Οι γενικές πληροφορίες για τις συνδρομές παρατίθενται στο http://deltahacker.gr/pdf12order και από την ίδια σελίδα γίνονται οι παραγγελίες συνδρομών.



MORE COLORS AVAILABLE
deltaHacker Field Bag
€25.50
deltaHacker Mugs
€12.50
deltaHacker Thermos® Can Cooler
€17.50
QUESTIONS?
ORDER BY PHONE?
CALL 1-877-809-1659
Where's My Stuff?
Track your recent orders.
View or change your order.
Shipping & Returns
See our shipping rates.
Make a return or exchange.
Need Help?
Satisfaction Guaranteed.
Visit our Customer Service.
is powered by CafePress Inc. Copyright © 1999-2015 CafePress.com. All rights reserved.
olicy | Trademark & Copyright Information
urrency rates are updated daily and may fluctuate.
Δώρα με στιλ, που έχουν *αληθινή* αξία.
Όλα στο μαγαζάκι του deltaHacker,
κάτω από το Cafepress.
Περάστε μια βόλτα από το
http://www.cafepress.co.uk/deltahacker
–σας περιμένουμε!

Πράγματα που μ' αρέσουν

του Χρήστου Βαρελά

Αγαπητές κι αγαπητοί,

- Σήμερα το μεσημέρι έλαβα email από συνδρομητή, ο οποίος ευγενικά μου ζήτησε ένα invitation για το Keybase. Δεν είναι το πρώτο email για το Keybase που έχω πάρει αυτές τις μέρες – κι αφορμή έχει σταθεί το σχετικό άρθρο που μπορείτε να διαβάσετε στο παρόν τεύχος (https://deltahacker.gr/?p=15567).
- Ας μην επαναλαμβάνω κι εδώ σε τι αφορά το Keybase. Θα πω μόνο ότι προς το παρόν η υπηρεσία είναι σε πρώιμο δοκιμαστικό στάδιο και για να συμμετάσχεις χρειάζεται να σε προσκαλέσει ένας ήδη εγγεγραμμένος χρήστης. Όταν το Keybase αρχίσει να λειτουργεί κανονικά και είναι ανοικτό για όλους, σκέφτομαι πως θα έχει καλές πιθανότητες να δώσει καλή ώθηση στη διάδοση των τεχνικών κρυπτογραφίας δημοσίου κλειδιού. Έτσι, όποιοι επιθυμούν να κρυπτογραφούν ισχυρά, π.χ., το email τους (https://deltahacker.gr/?p=14213), θα είναι σε θέση να το κάνουν πολύ πιο εύκολα.
- Ή τουλάχιστον αυτό ελπίζουν οι άνθρωποι του πρότζεκτ.
- Μη φανταστείτε ότι με πιάνει κάποια ιδιαίτερη τρέλα με την ανωνυμία ή/και με την ιδιωτικότητα. Κάθε άλλο, δεν ζω με το φόβο της διαρκούς κι αδιάκριτης παρακολούθησης από σκοτεινές δυνάμεις πέρα και πάνω από εμένα, ούτε αγχώνομαι με το ενδεχόμενο της υποκλοπής ευαίσθητων δεδομένων που με αφορούν. Απλά, θεωρώ ότι όποιος επιθυμεί να προστατεύει την ανωνυμία ή/και την ιδιωτικότητά του, πρέπει να έχει ανεμπόδιστη πρόσβαση στα κατάλληλα εργαλεία για να το πετυχαίνει.
- Ένας από τους λόγους που χαίρομαι με το Keybase, είναι διότι εδώ και μήνες προσπαθώ να κρυπτογραφώ το email μου καθημερινά. Κι επειδή η κρυπτογράφηση του email χρειάζεται δύο για να λειτουργήσει, όταν ο άλλος δεν έχει το δημόσιο πιστοποιητικό μου ή δεν ενδιαφέρεται ή απλά βαριέται να ψάξει το όλο θέμα, ε, τουλάχιστον έχω κανονίσει ώστε όλα τα emails που στέλνω να υπογράφονται ψηφιακά, αυτόματα.
- Αρκετοί συνδρομητές του περιοδικού ενδιαφέρονται για την κρυπτογραφία δημοσίου κλειδιού. Λιγότεροι, ομολογουμένως, εφαρμόζουν τις σχετικές τεχνικές στην πράξη, καθημερινά. Τώρα με το Keybase, ίσως παρακινηθούν κι ασχοληθούν περισσότεροι.

- Μ' αρέσει αυτή η σκέψη.
- Εκτός από την κρυπτογράφηση –ή έστω την ψηφιακή υπογραφή– του email, αυτό που επίσης μ' αρέσει είναι ότι χρησιμοποιώ ολοένα και πιο αραιά τους mail servers των μεγάλων, δηλαδή των Google, Yahoo, Microsoft κ.λπ. Αντίθετα, έχω εδώ και κάτι μήνες τον δικό μου mail server (https://deltahacker.gr/?p=14096) και είμαι στην ευχάριστη θέση ν' αναφέρω ότι λειτουργεί απείρως πιο αξιόπιστα απ' όσο περίμενα. Συνεχίζω όμως να χρησιμοποιώ και το Gmail. Το email για το deltaHacker.gr είναι ακόμα hosted από την Google και πρέπει να ομολογήσω ότι διστάζω να το μεταφέρω στο Mail-in-a-Box (εν συντομία MiaB – σ' αυτό βασίζεται ο DIY mail server μου). Αντίγραφα εφεδρείας παίρνω, όμως η αλληλογραφία για το περιοδικό είναι εξαιρετικά κρίσιμη για να μην την εμπιστεύομαι στα έμπειρα χέρια της Google.
- Αυτό το προηγούμενο, με τα έμπειρα χέρια, στην πραγματικότητα σημαίνει ότι αφενός φοβάμαι τον εαυτό μου, αφετέρου έχω μια μόνιμη ανησυχία για το μέλλον του MiaB. Εξηγούμαι: Για έναν DIY mail server, ο μόνος υπεύθυνος για την εύρυθμη λειτουργία του είναι ο ίδιος ο κάτοχός του. Με τις ιδέες που μου 'ρχονται κατά καιρούς, ικανό μ' έχω να τα κάνω μαντάρα ένα πρωί (εντάξει, μπορεί να 'ναι και βράδυ). Επίσης: Είμαι βέβαιος, όπως βέβαιος είναι κι οποιοσδήποτε άλλος στον πλανήτη ασχολείται έστω και λίγο με την τεχνολογία, ότι η Google θα μας συντροφεύει για πολλά πολλά χρόνια ακόμα. Για τον αξιέπαινο πρωτεργάτη του MiaB, δεν είμαι τόσο σίγουρος. Όχι ότι πρόκειται να πάθει κάτι ο άνθρωπος, δηλαδή. Απλά, κάποια στιγμή ίσως αποφασίσει ότι δεν θέλει ή δεν μπορεί ν' ασχοληθεί άλλο με το πρότζεκτ. Το MiaB βέβαια είναι Open Source, οπότε μετά από ενδεχόμενη απόσυρση του εμπνευστή του μάλλον θα πάρει άλλος τα ηνία. Αλλά μπορεί και όχι. Ξέρω κι εγώ;
- Μήπως ανησυχώ υπερβολικά κι άδικα; Μπορεί, μπορεί. Σε κάθε περίπτωση, η δυνατότητα για εύκολη μεταφορά των emails από mail server σε mail server με ησυχάζει κάπως. Τώρα που σας γράφω τις ανησυχίες μου και το σκέφτομαι πιο ψύχραιμα, μπορεί τελικά να το κάνω το μεγάλο βήμα και να φέρω *όλο* το email μου στην πλατφόρμα του MiaB και να απεμπλακώ από την Google.
- Θα το σκεφτώ λίγο ακόμα και θα δω.
- Εκτός από την αυτονομία που σου δίνει το MiaB, έχει και τον δικό του name server. Ο χειρισμός του είναι πανεύκολος και την ίδια στιγμή είναι πανίσχυρος. Παρέχει επίσης υπηρεσίες dynamic DNS κι έτσι έχω φροντίσει ώστε να φτάνω στο οικιακό δίκτυο εύκολα, χωρίς να έχω ιδέα για το δυναμικό IP που έχει δώσει ο ISP στον router μου. Ο name server του MiaB με παρακίνησε να

ξεσκονίσω και τις γνώσεις μου περί DNS. Μετά μάλιστα στρώθηκα κι ετοίμασα σχετική σειρά άρθρων για το περιοδικό (https://deltahacker.gr/?p=14712). Μιλάμε για τρομερή διασκέδαση – και το εννοώ.

- Λοιπόν, θα πω κι ένα τελευταίο για το MiaB και μετά θα σας αφήσω να χαρείτε το τεύχος.
- Έχεις αγοράσει, λοιπόν, ένα νέο domain – ας πούμε το colder.xyz. Κάποια στιγμή στήνεις κι εγκαθιστάς το MiaB σε ένα ωραιότατο VPS, ώστε να χειρίζεται το email για το συγκεκριμένο domain. Όλα καλά. Μαζί με τον mail server, ως δώρο παίρνεις κι έναν name server που είναι authoritative για το colder.xyz. Άριστα. Ξέρετε κάτι; Σαν να μην έφτανε το δώρο του name server, έχεις και web server για ένα static site που, προαιρετικά, μπορείς να σερβίρεις από το ίδιο VPS.
- Στην αρχή αγνόησα το θέμα με το static site. Σύντομα όμως η ιδέα με κέρδισε. Είναι που κατά καιρούς με πιάνει η όρεξη να γράφω τα δικά μου, οπότε ήθελα ένα απλό, απέριττο και ταχύτατο blog όπου θ' ανεβάζω τα πονήματά μου (κυρίως περί ανέμων και υδάτων).
- Κάπως έτσι, στο VPS που έχω το MiaB με τον name server του, κάποια στιγμή πήρε σάρκα και οστά και το https://colder.xyz. Το static site generator πίσω από το blog είναι το ταχύτατο Hugo. Είναι πολλά ακόμα που θέλω να μάθω για το Hugo, από νωρίς όμως άρχισα να παίζω –στο χαλαρό– και στο πλαίσιο των πειραματισμών μου έχω ανεβάσει και μερικά posts.
- Ορισμένοι φίλοι κι αναγνώστες από νωρίς φρόντισαν κι έδειξαν το ενδιαφέρον τους για τους static site generators. Να πω την αλήθεια, περίμενα ότι θα συνέβαινε κάτι τέτοιο. Πολλοί εκ των αναγνωστών μας θέλουν να εστιάζουν στην ουσία, οπότε από το blog τους ενδιαφέρονται κυρίως για το περιεχόμενο κι όχι για τις σφυρίχτρες και τα κουδούνια, όπως λέει κι ο κος Αρχισυντάκτης (TM).
- Τώρα που το τεύχος 050 του deltaHacker είναι έτοιμο και η επόμενη εβδομάδα θα είναι λίγο πιο χαλαρή, λέω να επιτεθώ στο Hugo. Αν το κάνω –που μάλλον θα το κάνω– θα ετοιμάσω κι ένα άρθρο για το τεύχος 051. Ή, όπως συχνά συμβαίνει, αντί για ένα άρθρο μπορεί να ετοιμάσω μίνι αφιέρωμα (διάβαζε: τουλάχιστον δύο άρθρα).
- Θα δείξει.



σνέιλ μέιλ

Αγαπητοί μου,

Σήμερα λοιπόν μου μπήκε η ιδέα να σας ετοιμάσω πρωτοχρονιάτικο Καζαμία.

Την ιδέα μου την έβαλε στην πραγματικότητα η Κυρία Καίτη, βλέποντάς με να βγάζω την γραφομηχανή. "Καζαμία θα γράψεις;" με ρωτάει.

"Καζαμία γράφει;" ακούγεται από την κουζίνα η μητέρα της Εξαδέλφης.

"Ναι, τώρα, όλα τα είχαμε, ο Καζαμίας μας έλειπε" ακούστηκε από κάπου μέσα η Εξαδέλφη.

Ξαφνικά όλοι αρχίσαν να μιλάνε για τον Καζαμία. Οπότε μου ήρθε έμπνευση.

Μόνο ο ανιψιός παρέμενε σιωπηλός. Τον πρόσεξα με την άκρη του ματιού να τσιμπάει ένα ακόμα μελομακάρουνο και να συνεχίζει να διαβάζει κάτι στο τάμπλετ του.

Γενικώς για κάποιον λόγο σήμερα είχαμε πολυκοσμία στο σπίτι. Κατ' αρχάς, οι καθ' ύλην αρμόδιοι ήταν στην κουζίνα και ετοίμαζαν την βασιλόπιτα.

Μετά είχαμε τον ανιψιό, να εξολοθρεύει τα μελομακάρουνα.

Προσπαθούσαμε να τον συνετίσουμε, η Εξαδέλφη κι εγώ. Του φωνάζαμε. "Άσε καλέ κανένα μελομακάρουνο για τους άλλους."

"Καλέ μην τρως μελομακάρουνα", πεταγόταν κι η Μαιρούλα. "Παχαίνει."

"Σιγά μην παχαίνει."

"Παχαίνει, σου λέω, αφού το έγραφε στο ίντερνετ."

Τέλος πάντων, να μην σας τα πολυλογώ, όλοι μέσ' στο εορταστικό πνεύμα. Το κουδούνι να χτυπάει κάθε λίγο και λιγάκι για να μας πούνε τα κάλαντα, τα μελομακάρουνα να λιγοστεύουν με ραγδαίους ρυθμούς, και οι άλλοι στην κουζίνα, οι καθ' ύλην αρμόδιοι όπως είπα, να αναρωτιούνται τι μπαχαρικά να βάλουν στην βασιλόπιτα ώστε να είναι, μεν, παραδοσιακή, αλλά να έχει και λίγο χαρακτήρα.

Κι εγώ να σκέφτομαι τον Καζαμία.

Για να πάρω ιδέες πήγα στις κούτες που έχουμε στο τρίτο πατάρι και άρχισα να ψαχουλεύω.

Έψαχνα κάτι παλιά τεύχη εκείνου του παλιού του περιοδικού στο οποίο δούλευα μέχρι πριν πέντε-τόσα χρόνια.

Πάνε κιόλας πέντε χρόνια! Άκουσον άκουσον. Τότε που αποκομμένος στο Πόρτο Γερμενό, λόγω των ακινητοποιημένων ταξιών και των άλλων κινητοποιημένων επαγγελματικών κλάδων, είχα αποφασίσει να κινητοποιηθώ κι εγώ, μεταφράζοντας τη θερινή μου ραστώνη ως ένδειξη αγωνιστικότητας και διεκδικητικού πνεύματος.

Τέλος πάντων, νόμος είναι το δίκιο του εργάτη, που έλεγαν προ ημερών στην ΕΒΓΑ του κυρίου Κώστα.

Λοιπόν, βρίσκω που λέτε τα τεύχη που έψαχνα, κι έκατσα κατάχαμα να τα ξεφυλλίζω.

Τεύχος Ιανουαρίου 2000. Είχανε βάλει τότε έναν Καζαμία, και για κάποιο λόγο το θυμόμουν. Έκατσα να τον διαβάσω όλον. Τον υπέγραφαν από κοινού ένας άγνωστός μου (τότε) συνεργάτης (που εκ των υστέρων έμαθα

πως σχετίζεται με τον Κύριο Εκδότη: έχουν το ίδιο όνομα) και κάποιος άλλος τον οποίο αγνοούσα τότε, και συνεχίζω να αγνοώ ακόμα. Ενεργητικά.

Επρόκειτο για Καζαμία ολόκληρου του 21ου αιώνα.

Αντιγράφω αποσπάσματα από κάποιες απ' τις προβλέψεις του:

2002, Απρίλιος: *Εκτοξεύεται ο δορυφόρος KEO, ο οποίος είναι προγραμματισμένος να κάνει γύρους στο διάστημα και να επιστρέψει στη Γη μετά από 50.000 χρόνια. Μέσα στον δορυφόρο υπάρχει πλήθος πληροφοριών για την σημερινή Γη, τους κατοίκους της και τον πολιτισμό τους.*

2005, Νοέμβριος: *Επιστήμονες της IBM κατασκευάζουν τον "WiseGuy", ένα πρωτότυπο νευρωνικό επεξεργαστή που χρησιμοποιεί ένα επαναστατικό υπολογιστικό μοντέλο. Εκπρόσωπος τύπου της IBM, εξηγεί πως "από τη στιγμή που του δίνεται ένα πρόβλημα, ο WiseGuy προσποιείται ότι δεν μπορεί να το λύσει, ταυτόχρονα όμως το εξετάζει από πολλές πλευρές, διατηρώντας πάντα μια απόσταση ασφαλείας. Τελικά, το πρόβλημα κουράζεται και παραδίνεται".*

2012, Οκτώβριος: *Ολοκληρώνεται με επιτυχία η αποστολή ειδικού δειγματοληπτικού διαστημοπλοίου στον Άρη, η συλλογή δειγμάτων από το έδαφος και την ατμόσφαιρα, και η επιστροφή του σκάφους στη Γη. Οι επιστήμονες ελπίζουν να απαντήσουν σε πολλά από τα ερωτήματά τους για τον ερυθρό πλανήτη, κυρίως για το αν υπήρξε κάποια στιγμή νερό στην επιφάνειά του.*

2016: *Οι Ολυμπιακοί Αγώνες της χρονιάς αυτής διοργανώνονται στο Internet. Πρόκειται για τους πρώτους πραγματικά διαδραστικούς αγώνες. Ο καθένας μπορεί να "ζήσει" την εμπειρία οποιουδήποτε αγωνίσματος δοκιμάζοντας τις δυνάμεις του σε περιβάλλοντα εικονικής πραγματικότητας.*

Έχει πλάκα να τσεκάρεις αναδρομικά παλαιότερες προβλέψεις για το μέλλον. Η πλάκα δεν έγκειται τόσο στο πόσο μέσα ή όχι έχουν πέσει οι προβλέψεις, ως τέτοιες, αλλά στην σύγκρισή τους με το τι συνέβη στην πραγματικότητα κατά τις στιγμές στις οποίες αναφέρονταν (οι προβλέψεις).

Να, πάρτε, για παράδειγμα την πρόβλεψη για τον Δορυφόρο KEO. Ίσως να μην το γνωρίζετε, αλλά πράγματι σκέψεις για έναν τέτοιο δορυφόρο υπάρχουν. Μόνο που δεν εκτοξεύτηκε τον Απρίλιο του 2002. Δεν έχει εκτοξευτεί ακόμα. Οι πιο πρόσφατοι σχεδιασμοί τοποθετούν την εκτόξευση στο 2017, με αναμενόμενη επιστροφή πράγματι σε 50.000 χρόνια. Πενήντα χιλιάδες χρονάκια. Ποιος ζει ποιος πεθαίνει μέχρι τότε.

Κατά τα άλλα, Απρίλιο του 2002 είχε μόλις ξεκινήσει ο δεύτερος πόλεμος του Κόλπου. Είχε μεσολαβήσει η 11η Σεπτεμβρίου, και λίγο μετά την 11η Σεπτεμβρίου είχε ξεκινήσει η επέμβαση στο Αφγανιστάν.

Ωραία. Δορυφόρο σου ήθελε ο άλλος.

Άλλη πρόβλεψη: Νοέμβριος 2005, η IBM φτιάχνει τον WiseGuy, που απαντάει σε προβλήματα ταλαιπωρώντας τα μέχρι να παραδώσουν τα όπλα. Χιουμοριστική μεν η πρόβλεψη, αλλά ενδεικτική.

Πλάκα πλάκα, η IBM έφτιαξε πράγματι έναν τέτοιο υπολογιστή, μόνο που τον είπανε Watson και όχι WiseGuy. Το 2006 βρισκόταν ακόμα στις δοκιμές. Το 2011 μπόρεσε να παίξει στο Jeopardy και να νικήσει τους ανθρώπινους αντιπάλους του χωρίς πρόβλημα.

Πρόβλεψη: Οκτώβριος 2012. Αποστολή στον Άρη, συλλογή δειγμάτων, επιστροφή στην Γη.

ΟΚ, αποστολές στον Άρη έχουν πλέον γίνει αρκετές, αλλά επί του παρόντος δεν έχει ακόμα λυθεί το πρόβλημα τις επιστροφής. Μόνο στο σινεμά. Όμως το ερώτημα για το νερό στον Άρη έχει πια απαντηθεί: Φαίνεται ότι συνεχίζει να υπάρχει νερό ακόμα και σήμερα.



Όσον αφορά τους Ολυμπιακούς του '16, ο Καζαμίας του Ιανουαρίου 2000, σοφά ποιώντας, δεν διακινδυνεύει να μας πει πού θα γίνουν. Δεν θα μπορούσε να φανταστεί τη Βραζιλία. Αλλά μας λέει πως θα γίνουν και στο ίντερνετ.

Τώρα, για αγώνες στο ίντερνετ, χλωμό το βλέπω, και όχι μόνο λόγω τεχνολογίας. Αλλά διαβάζω για Βραζιλία και βλέπω πως η οικονομία της χώρας είναι στα πρόθυρα της αβύσ-

σου. Άκουγα προ ημερών πως η πρόεδρος Ντίλμα Ρούσεφ το παλεύει δεν το παλεύει να σταθεί στο ύψος των προ έτους προεκλογικών της υποσχέσεων που --υποσχέσεις που της διασφάλισαν μια δεύτερη θητεία. Οι Ολυμπιακοί φαντάζουν πλέον ως μια δυσάρεστη δέσμευση.

Μια που λέμε για προεκλογικές υποσχέσεις, θυμήθηκα κάτι άλλο.

Η άλλη ιδέα που είχα --εναλλακτική του Καζαμία-- ήταν να μιλούσα για τη χρονιά που μας πέρασε. Δίκην ανασκόπησης.

Αποφάσισα να μην την κάνω.

Ή μάλλον, για να το πω πιο συγκεκριμένα: Αποφάσισα να μην κάνω ανασκόπηση επί τη βάσει συγκεκριμένων γεγονότων.

Για διάφορους λόγους -- εκ των οποίων ο σημαντικότερος έχει να κάνει με τη δυσκολία μας ως παρατηρητών να αποφασίσουμε τι συνιστά γεγονός και τι όχι.

Να μου πείτε, αυτά τα έχω ξαναπεί.

"Λυπάμαι που σας το χαλάω", έγραφα το Σεπτέμβριο του 2014, τεύχος 36, "αλλά καθαρά γεγονότα, ως τέτοια, *δεν υπάρχουν*. Υπάρχουν μόνο ερμηνείες. Και κάθε ερμηνεία εμπεριέχει ήδη μιαν άποψη για το πώς έχουν τα πράγματα. Κάθε ερμηνεία είναι ήδη μια θεωρία (υπό την έννοια ότι εντάσσεται ήδη σε ένα πλαίσιο κατανόησης του πεδίου στο οποίο λαμβάνει χώρα το φαινόμενο)."

OK. Nothing new. Ανθρώπινος περιορισμός και τα σκυλιά δεμένα.

Θα περίμενα, λοιπόν, κάτι. Θα περίμενα να έχουμε πλέον κάποια επίγνωση αυτού του περιορισμού, και να τον θυμόμαστε όταν ετοιμαζόμαστε να κατακεραυνώσουμε το συνομιλητή μας κάθε φορά που διαφωνούμε.

Θα περίμενα, ας πούμε, και θα ήλπιζα, να είμαστε λιγότερο έτοιμοι να χαρακτηρίσουμε τον άλλο ως άσχετο / νεοφιλελέ / προσκυνημένο / γερμανοτσολιά / μνημονιακό, κ.λπ., όταν λέει κάτι που δεν μας αρέσει ή μας φαίνεται λάθος.

Αγαπητοί μου,

Ο τρόπος με τον οποίο έγινε ο δημόσιος διάλογος μέσα στο τελευταίο δωδεκάμηνο με τάραξε, με απογοήτευσε, και με πλήγωσε βαθύτατα. Προσωπικά το λέω. Προσωπικά το εννοώ.

Με πλήγωσε άσχημα. Ούτε μπορώ, ούτε θέλω να το κρύψω. Άλλωστε ακόμα δεν το έχω ξεπεράσει.

Η διαφωνία αγαπητοί μου είναι μέσα στο παιχνίδι. Ακόμα και η έντονη διαφωνία. Αλλά άλλο διαφωνία και άλλο να βγαίνει ο άλλος και να σε προσβάλλει πως είσαι αφελής ή κάνεις τον αφελή όταν λες τη γνώμη σου για το "ηρωικό" δημοψήφισμα και η γνώμη σου είναι διαφορετική από τη δική του.

Άλλο διαφωνία κι άλλο να σε περιφρονεί ο άλλος με τη σιγουριά των δέκα σοφών της αρχαιότητας, όταν δεν ερμηνεύεις με το δικό του τρόπο το τάδε ή το δείνα "αντικειμενικό" και "αδιαμφισβήτητο" συμβάν.

Άλλο διαφωνία κι άλλο να μην κάνει ο άλλος καν τον κόπο να σου απαντήσει σε κάτι που του λες...

Αυτά δεν είναι διαφωνίες, αγαπητοί μου. Αυτά είναι bullying. Bullying εξ αριστερών.

Λοιπόν, ένα πράγμα με δίδαξε το 2015, που λέτε. Με δίδαξε πόσο εύκολα περνάς από το ένα στο άλλο: από τη συζήτηση στο bullying.

Τώρα θα μου πείτε, το χρειαζόμουν αυτό το μάθημα; Δεν νομίζω.

Τέλος πάντων. Ό,τι έγινε, έγινε. Ας πάει στο καλό.

Σας εύχομαι καλή χρονιά.

Σας ασπάζομαι,
Θείος Ακάκιος



e-card
4000 1234 5678 9010
4000
GOOD THRU
MONTH / YEAR
12 / 19
MasterCard
Cardholder Name
Κάρτα e-coin, που δέχεται Bitcoins: Δουλεύει όπως όλες οι κανονικές κάρτες Debit, παντού στον κόσμο και για online αγορές.
Αποκτήστε τη δική σας με κλικ εδώ
και κερδίστε αυτομάτως
25% έκπτωση
κατά την παραγγελία της κάρτας.

Skill: Intermediate
Tags: SSTV, Raspberry Pi, SDR, scanline, RPiTX, VIS

SSTV ...ή αλλιώς άκου να δεις!

Την εποχή του Markoni, την επανάσταση στον τομέα της μεταφοράς πληροφοριών την έφεραν οι ραδιοεπικοινωνίες. Όπως πολλές άλλες τεχνολογίες, έτσι κι αυτή ξεκίνησε από εργαστήρια με δυσεύρετο και πανάκριβο εξοπλισμό αλλά τελικά διαδόθηκε σε ένα μεγάλο μέρος του κοινού. Οι hackers εκείνης της εποχής, οι ραδιοερασιτέχνες, αξιοποιούσαν τα ραδιοκύματα για να εκπέμψουν δικά τους μηνύματα. Σήμερα, η νέα γενιά ρεδιοερασιτεχνών χρησιμοποιεί τα ραδιοκύματα για να εκπέμψει ήχο, εικόνα αλλά *και* δεδομένα.

του Πέτρου Κυλαδίτη

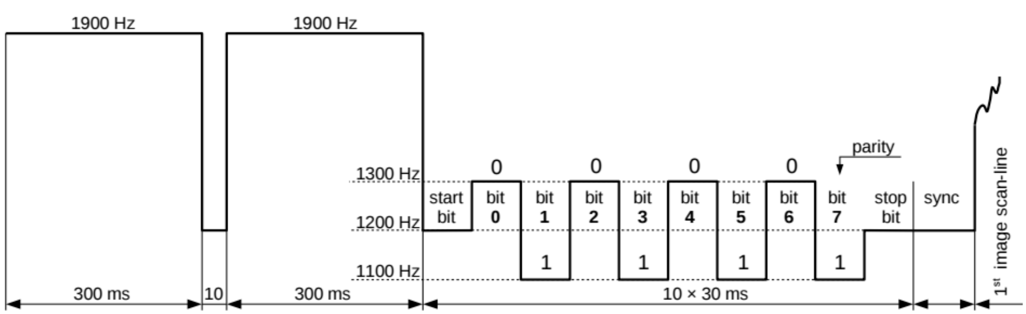
Ήδη από την εισαγωγή του άρθρου είμαστε σίγουροι ότι έχετε καταλάβει πού το πάμε. Η μετάδοση σκέτου ήχου είναι πολύ μπανάλ πλέον. Σε ό,τι αφορά στη μετάδοση της εικόνας υπάρχει ένα ευρέως διαδεδομένο πρότυπο, που ακούει στο όνομα SSTV (Slow-Scan TeleVision). Μη σας μπερδεύει όμως η παρουσίας της λέξης Television, μέσα στο όνομα. Δεν πρόκειται για τηλεοπτικό σήμα σαν αυτό που μεταφέρουν τα πρότυπα PAL, NTSC και SECAM, των 25 με 30 καρέ ανά δευτερόλεπτο. Μέσω του SSTV μπορούμε να μεταδώσουμε αναλογική εικόνα (ασπρόμαυρη ή έγχρωμη) με ρυθμό ανανέωσης από 10 έως 1/114 FPS, ανάλογα με τον τρόπο μετάδοσης, τα χρώματα και την ανάλυση. Μήπως αναρωτιέστε σε τι μπορεί να φανεί χρήσιμη μια τόσο αργή μετάδοση; Οι εξαιρετικά χαμηλές προδιαγραφές του προτύπου επιτρέπουν τη μετάδοση σε πολύ μεγάλες αποστάσεις και μέσα από διαύλους μικρού εύρους, που προορίζονται για ήχο. Σκεφτείτε ότι το SSTV αξιοποιήθηκε σε διαστημικά προγράμματα όπως το Luna 3, το Faith 7 και το Apollo. Μπορεί να μη σας πέρασε ποτέ από το μυαλό, αλλά το πασίγνωστο βίντεο της βόλτας του Neil Armstrong στη Σελήνη, που προβλήθηκε ζωντανά σε ολόκληρο τον κόσμο, μεταδόθηκε από τη Σελήνη προς τον επίγειο σταθμό μέσω SSTV. Και για όσους έχουν καλπάζουσα φαντασία, ναι, ο διαστημικός σταθμός ISS εκπέμπει ακόμα και σήμερα σήμα SSTV.

Το SSTV είναι αναλογικό σήμα που μεταφέρει ψηφιοποιημένα δεδομένα. Έτσι, πριν από την αποστολή απαιτείται μια διαδικασία κωδικοποίησης των ψηφιακών δεδομένων και, αντίστοιχα, μετά τη λήψη του αναλογικού σήματος ακολουθεί μια διαδικασία αποκωδικοποίησης. Όταν αναφερθήκαμε στο ρυθμό ανανέωσης της εικόνας, είπαμε ότι δεν είναι σταθερός κι ότι εξαρτάται από διάφορους παράγοντες.

Ε, λοιπόν, ένας από αυτούς είναι η κωδικοποίηση. Στον κόσμο του SSTV, βλέπετε, δεν υπάρχει κάποιο de facto πρότυπο για την ενσωμάτωση των δεδομένων της εικόνας στα ραδιοκύματα. Υπάρχουν πολλά διαφορετικά πρότυπα, με το καθένα να παρουσιάζει τα δικά του προτερήματα και μειονεκτήματα. Τα δημοφιλέστερα εξ' αυτών, που εξακολουθούν να χρησιμοποιούνται ως τις μέρες μας, είναι τα Martin, Scottie, Robot και οι παραλλαγές τους.

Στα ενδότερα του SSTV

Τώρα ενδέχεται να προβληματιστήκατε: Πώς μπορεί ο αποδέκτης ενός σήματος να γνωρίζει το πρότυπο κωδικοποίησης που χρησιμοποιεί ο αποστολέας; Μια απλοϊκή λύση θα ήταν να έχει προηγηθεί συνεννόηση μεταξύ των δύο και να γίνεται χρήση ενός προσυμφωνημένου προτύπου. Κάτι τέτοιο όμως δεν είναι πάντα εφικτό και σίγουρα δεν είναι καθόλου βολικό. Το ίδιο πρόβλημα εμφανίζεται και σε άλλα συστήματα επικοινωνίας. Σκεφτείτε για παράδειγμα τα FAX ή τα παλιά MODEM, που επέτρεπαν τη σύνδεση στο Internet μέσω των αναλογικών τηλεφωνικών γραμμών. Σε αυτές στις συσκευές, πριν από τη μετάδοση των "ωφέλιμων" δεδομένων προηγείται ένα στάδιο διερεύνησης των υποστηριζόμενων προτύπων. Αυτή η διαδικασία διερεύνησης καταλήγει σε μια "συμφωνία" μεταξύ των συσκευών, για το πώς ακριβώς θα ανταλλάσουν δεδομένα. Αυτό το αρχικό στάδιο της επικοινωνίας είναι γνωστό ως χειραψία ή handshake, όπως συνηθίζουν να λένε στην ευρύτερη περιοχή της Καλλιμασιάς Χίου (Σ.τ.Ε. Κωνσταντίνε;!) Το SSTV βέβαια δεν προβλέπει την αμφίδρομη επικοινωνία και δεν θα μπορούσε να πραγματοποιηθεί κάποιο handshake, ούτε θα πετυχαίναμε κάποια συμφωνία. Εδώ ακολουθείται μια απλούστερη προσέγγιση: Το σήμα που εκπέμπει ο αποστολέας, μαζί με τα δεδομένα της εικόνας μεταφέρει και το χρησιμοποιούμενο πρότυπο κωδικοποίησης. Ο δέκτης λαμβάνει αυτή την πληροφορία και το σύστημα αποκωδικοποίησης ρυθμίζεται αυτομάτως. Ωραία, θα πείτε, αλλά τι γίνεται αν ο δέκτης καθυστερήσει να συνδεθεί και χάσει την αρχή του σήματος; Κανένα πρόβλημα! Το επιλεγμένο πρότυπο κωδικοποίησης εκπέμπεται διαρκώς, όχι μόνο στην έναρξη της εκπομπής.



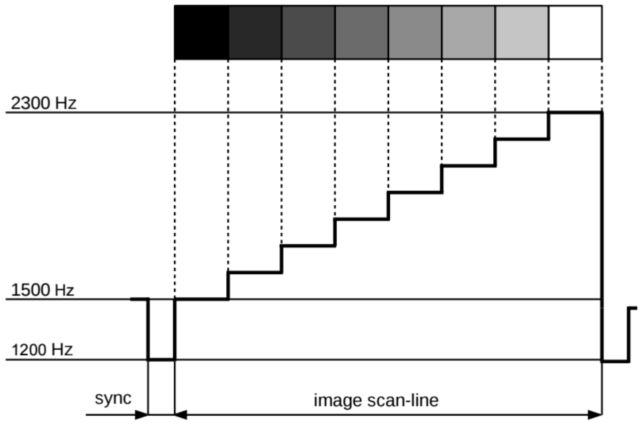
Στην εικόνα βλέπουμε το περίγραμμα (envelope ή contour) ενός σήματος SSTV και, συγκεκριμένα, του τμήματος με το VIS header. Το συγκεκριμένο ενσωματώνει τον κωδικό 42.

Κάθε φορά που ξεκινά η εκπομπή ενός καρέ (frame) της εικόνας, το πρότυπο SSTV προβλέπει την εκπομπή ενός σήματος με το όνομα VIS (Vertical Interval Signaling). Ουσιαστικά είναι κάτι σαν το Vertical Sync του VGA. Μόνο που στην περίπτωση του SSTV δεν πρόκειται για έναν απλό παλμό, αλλά για ένα ελαφρώς πιο σύνθετο σήμα. Πριν από την έναρξη του VIS εκπέμπονται τρεις τόνοι, ένας στα 1900Hz, ένας στα 1200Hz κι ένας ακόμα στα 1900Hz. Η αποστολή τους είναι εξαιρετικά απλή: Ειδοποιούν τον εκάστοτε δέκτη ότι ακολουθεί η εκπομπή του VIS. Το συγκεκριμένο σήμα, βλέπετε, έχει βαρύνουσα σημασία. Εκτός του ότι σηματοδοτεί την έναρξη ενός καρέ, περιλαμβάνει την πληροφορία για το χρησιμοποιούμενο πρότυπο κωδικοποίησης. Η διάρκεια του σήματος VIS ανέρχεται στα 30ms και στα δύο του άκρα (αρχή και τέλος) περιλαμβάνει δύο τόνους στα 1200Hz, με τον καθένα να διαρκεί 3ms. Ουσιαστικά, αυτοί οι δύο τόνοι αποτελούν το start bit και το stop bit αντίστοιχα. Όπως υποψιάζεστε, ο χρόνος των 3ms είναι η διάρκεια εκπομπής για όλα τα bits. Επομένως, αν αφαιρέσουμε το χρόνο των start και stop bits από τα 30ms που διαρκεί το VIS, απομένει χρόνος για οκτώ ακόμα bits. Από αυτά, τα επτά πρώτα αναπαριστούν την ταυτότητα του χρησιμοποιούμενου προτύπου κωδικοποίησης, ενώ το τελευταίο χρησιμοποιείται για την ανίχνευση σφαλμάτων κατά τη μετάδοση (parity bit). Το λογικό 0 μεταφέρεται ως ένας τόνος των 1300Hz, ενώ το λογικό 1 ως ένας τόνος των 1100Hz. Οι 128 διαφορετικές τιμές που μπορούν να περιγράψουν τα επτά bits είναι περισσότερες από τα διαθέσιμα πρότυπα κωδικοποίησης.

Μετά από το stop bit του VIS ακολουθεί η αποστολή των γραμμών της εικόνας. Η έναρξη κάθε γραμμής σηματοδοτείται με την εκπομπή ενός τόνου στα 1200Hz. Μπορείτε να τον φαντάζεστε σαν το horizontal sync του VGA. Μετά από κάθε τέτοιο τόνο, εκπέμπονται τα δεδομένα για τα pixels της εκάστοτε γραμμής. Με λίγα λόγια, η εικόνα στο SSTV σχηματίζεται pixel προς pixel και γραμμή προς γραμμή (scan lines). Εν ολίγοις, η εικόνα μεταφέρεται όπως σε όλα τα πρότυπα αναλογικής μετάδοσης της εικόνας. Παρεμπιπτόντως, αν θέλετε να μάθετε περισσότερα γι' αυτά τα πρότυπα και ειδικά για το VGA, μπορείτε να διαβάσετε το άρθρο "Κατασκευάστε έναν υπολογιστή, μέρος 3: Pixel προς pixel" (τεύχος 048, http://deltahacker.gr/deltahacker048).

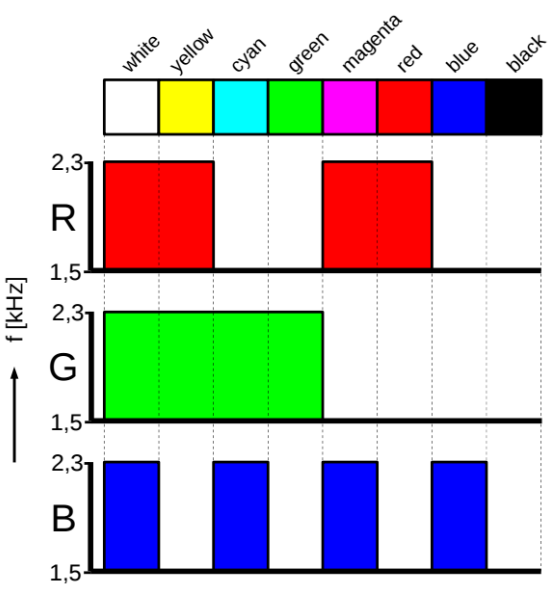
Κωδικοποίηση χρώματος

Το χρώμα των pixels αναπαρίσταται και πάλι με τη χρήση διαφορετικών συχνοτήτων. Για παράδειγμα, στο πρότυπο Robot B/W 24 η συχνότητα 1500Hz αντιστοιχεί σε ένα μαύρο pixel, ενώ η συχνότητα 2300Hz αντιστοιχεί σε ένα λευκό pixel. Οι ενδιάμεσες συχνότητες ερμηνεύονται σαν τις αντίστοιχες διαβαθμίσεις του γκρι. Κι επειδή το συγκεκριμένο πρότυπο προβλέπει σαν ελάχιστο βήμα για τη μεταβολή της συχνότητας τα 12,5Hz, στο διάστημα από 1500 έως 2300Hz μπορούν να "χωρέσουν" 64 αποχρώσεις του γκρι. Στο ίδιο πρότυπο, ο χρόνος που απαιτείται για τη σάρωση μιας γραμμής ανέρχεται στα 93ms, ενώ η ανάλυση της εικόνας είναι 320x240 (320 pixel ανά γραμμή και 240 γραμμές). Αν κάνουμε τις πράξεις και προσθέσουμε το χρόνο για τα σήματα οριζόντιου και κατακόρυφου συγχρονισμού, διαπιστώνουμε ότι η εκπομπή μιας ολόκληρης εικόνας διαρκεί περί τα 25 δευτερόλεπτα. Σας φαίνεται πολύ λίγο; Περιμένετε και θα δείτε.



Σε αυτό το διάγραμμα βλέπουμε πώς μεταβάλλεται το χρώμα των pixels, καθώς η συχνότητα του σήματος ανεβαίνει από τα 1500Hz στα 2300Hz. Τα σκαλοπατάκια οφείλονται στο γεγονός ότι το πρότυπο που έχει χρησιμοποιηθεί(RobotB/W)προβλέπει τη μεταβολή της συχνότητας μόνο κατά ένα συγκεκριμένο βήμα.

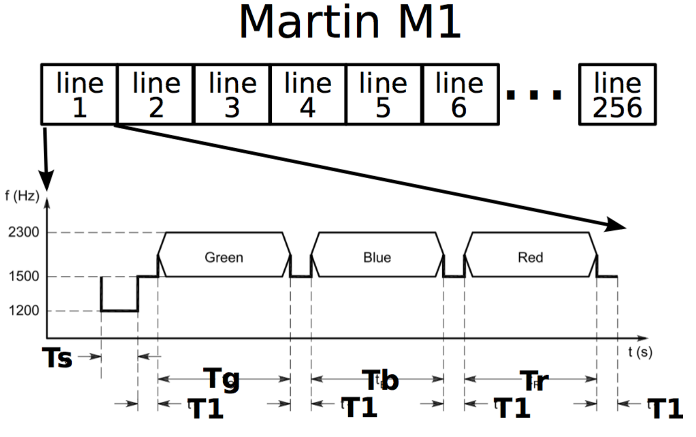
Στην περίπτωση των έγχρωμων προτύπων, όπως είναι το δημοφιλές Martin M1, για τον καθορισμό του χρώματος κάθε pixel συνδυάζονται τρεις τόνοι: Ένας για το κόκκινο, ένας για το πράσινο κι ένας για το μπλε. Για κάθε τόνο ισχύουν αυτά που περιγράψαμε νωρίτερα, στο πρότυπο Robot B/W 24. Επομένως, για κάθε χρωματική συνιστώσα έχουμε και πάλι 64 διαβαθμίσεις κι αυτό σημαίνει ότι το πλήθος των διαθέσιμων χρωμάτων ανέρχεται στα 64 x 64 x 64 = 256144 χρώματα. Σαν να λέμε, έχουμε 18 bits χρώματος – λίγο παράξενο βάθος, αλλά σίγουρα ικανοποιητικό. Οι τρεις τόνοι για κάθε συνιστώσα εκπέμπονται διαδοχικά (πρώτα για το πράσινο, μετά για το μπλε και τέλος για το κόκκινο). Η διάρκεια αυτών των τόνων είναι μεγαλύτερη από αυτή του προτύπου Robot, ενώ στο ενδιάμεσο παρεμβάλλονται και μικρές παύσεις. Τελικά, αν και η ανάλυση της εικόνας στο πρότυπο Martin M1 είναι μόλις 320x256, η εκπομπή μιας ολόκληρης εικόνας διαρκεί περί τα 2 λεπτά.



Στα έγχρωμα πρότυπα του SSTV, όπως είναι το Martin M1, για κάθε pixel εκπέμπονται τρεις διαφορετικοί τόνοι. Ο καθένας αντιστοιχεί σε ένα από τα τρία βασικά χρώματα (κόκκινο, πράσινο, μπλε). Εδώ βλέπουμε πώς συνδυάζονται οι τρεις τόνοι για να πάρουμε οκτώ κύρια χρώματα.

Κάθε πρότυπο υποστηρίζει διαφορετικό τρόπο κωδικοποίησης της οπτικής πληροφορίας και διαφορετική ανάλυση. Για παράδειγμα, αντί του κλασικού RGB, τα έγχρωμα πρότυπα Robot-κάτι (υπάρχουν αρκετές παραλλαγές) χρησιμοποιούν το χρωματικό φορμάτ YCrCb. Το Y περιγράφει τη φωτεινότητα, ενώ τα Cb και Cr περιγράφουν τις διαφορές στον χρωματικό τόνο σε σχέση με το μπλε και το κόκκινο αντίστοιχα. Άλλα πρότυπα, όπως το Wraase SC-2, προβλέπουν διπλάσιο χρόνο αποστολής για το πράσινο χρώμα. Αυτό σημαίνει ότι μπορούν να εκπεμφθούν περισσότερες αποχρώσεις. Η επιλογή του πράσινου δεν έγινε τυχαία και οφείλεται στο γεγονός ότι το ανθρώπινο μάτι είναι περισσότερο ευαίσθητο στο συγκεκριμένο χρώμα. Τέλος, κάποια άλλα πρότυπα στοχεύουν στην αποστολή ευκρινέστερης εικόνας, θυσιάζοντας το ρυθμό ανανέωσης. Για παράδειγμα, η εκπομπή μιας εικόνας στο Scottie DX ανέρχεται στα 4,5 λεπτά!

Κάθε πρότυπο κωδικοποίησης προβλέπει διαφορετικούς χρόνους για τη μετάδοση των "εξαρτημάτων" της εικόνας. Ωστόσο, οι γενικές αρχές παραμένουν ίδιες σε κάθε περίπτωση. Πριν εγκαταλείψουμε τη θεωρία θα θέλαμε να επιστήσουμε την προσοχή σας σε μια ενδιαφέρουσα λεπτομέρεια: Οι συχνότητες που χρησιμοποιούνται για τους διάφορους τόνους βρίσκονται κοντά στα 2KHz. Ξέρετε τι σημαίνει αυτό; Αν οδηγήσουμε το σήμα SSTV σε έναν ενισχυτή κι από εκεί σε ένα ηχείο, θα ακούσουμε καθαρά όλα τα επιμέρους σήματα που απαρτίζουν την εικόνα :D



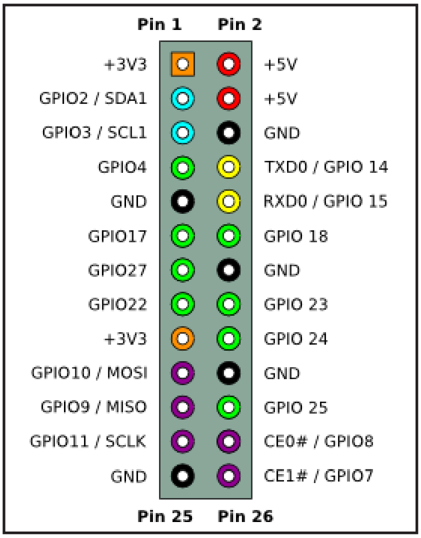
Η αναπαράσταση τμήματος ενός scanline, κωδικοποιημένο κατά Martin M1. Το μέγεθος Ts αντιστοιχεί στη διάρκεια του τόνου οριζόντιου συγχρονισμού, ενώ τα Tg, Tb και Tr αντιστοιχούν στις διάρκειες των τόνων για τα χρώματα πράσινο, μπλε και κόκκινο ενός pixel. Ο χρόνος T1 αποτελεί τις παύσεις που περιλαμβάνει το σήμα ανάμεσα σε κάθε τόνο.

Απαραίτητος εξοπλισμός

Όσοι σκέφτεστε ότι δεν έχετε τα τεχνικά μέσα για να παρακολουθήσετε τη συνέχεια του άρθρου, κάνετε λάθος. Ο εξοπλισμός που απαιτείται για την αποκωδικοποίηση σημάτων SSTV δεν είναι άλλος από τον υπολογιστή μας. Βέβαια για τη λήψη του σήματος απαιτείται εξοπλισμός SDR, αλλά και πάλι δεν πρόκειται για κάτι ακριβό ή σπάνιο. Μπορεί εκείνο το κύκλωμα τηλεόρασης σε στικάκι που έχετε ξεχασμένο σε κάποιο συρτάρι ή και η κάρτα τηλεόρασης που έχετε μόνιμα στον υπολογιστή, να είναι κατάλληλα. Περισσότερα για το SDR (Software-Defined Radio) μπορείτε να διαβάσετε στο ομώνυμο άρθρο του τεύχους 023 (http://deltahacker.gr/deltahacker023). Πάντως για την εκπομπή σημάτων SSTV τα

πράγματα είναι ακόμα απλούστερα. Προκειμένου να εκπέμψουμε σε μικρή εμβέλεια μπορούμε να στραφούμε στο αγαπημένο μας Raspberry Pi. Στη συνέχεια, για λόγους που εξυπηρετούν την παρουσίαση, θα εξετάσουμε τη μετάδοση σήματος SSTV και για το τέλος θα αφήσουμε τη λήψη και την αποκωδικοποίηση.

Οι παλιοί αναγνώστες θα θυμούνται το πρόγραμμα PiFM, με τη βοήθεια του οποίου ένας ακροδέκτης GPIO του RPi μετατρέπεται σε κεραία για τη μετάδοση σημάτων FM. Στο τεύχος 029 (http://deltahacker.gr/deltahacker029) αξιοποιήσαμε το εν λόγω εργαλείο (PiFM) για την αναμετάδοση του σήματος ενός ραδιοφωνικού σταθμού, ενώ στο τεύχος 037 (http://deltahacker.gr/deltahacker037) μπήκαμε στον πειρασμό να δημιουργήσουμε το δικό μας σύστημα μετάδοσης δεδομένων. Όσοι έχετε διαβάσει τα αντίστοιχα άρθρα, γνωρίζετε ότι το PiFM υποστηρίζει μόνο την αναμετάδοση ηχητικών δεδομένων, αποκλειστικά με διαμόρφωση FM και μόνο στο σχετικό φάσμα συχνοτήτων. Το PiFM, όμως, αποτελεί Ελεύθερο Λογισμικό κι ο πηγαίος κώδικάς του διανέμεται δωρεάν και χωρίς περιορισμούς. Μη νομίζετε ότι θα το ρίξουμε ξαφνικά στις προγραμματιστικές επεμβάσεις. Αν κάναμε κάτι τέτοιο θα έπρεπε να ξεχάσουμε για τα καλά το SSTV, μια που θα χρειαζόμασταν αρκετές σελίδες σε αρκετά τεύχη. Αυτή τη φορά, λοιπόν, θα εργαστούμε με μια τροποποιημένη εκδοχή του PiFM, που δημιούργησε ο Evariste Courjaud. Αναφερόμαστε στο εργαλείο RPiTX, το οποίο επιτρέπει τη μετάδοση με διαμόρφωση FM, AM, SSB, VFO, FSQ και SSTV, σε συχνότητες από 130KHz έως 750MHz. Το RPiTX, σε αντίθεση με το PiFM, δεν μπορεί να μεταδώσει "απλά" αρχεία ήχου, ούτε και "σκέτες" εικόνες. Υποστηρίζει μόνο αρχεία IQ και FT. Τα πρώτα είναι γνωστά σε όλους τους ραδιοερασιτέχνες που χρησιμοποιούν το GNU Radio, ενώ τα δεύτερα παράγονται από μια σειρά μικρών προγραμμάτων που συνοδεύουν το RPiTX. Όπως δηλώνουν τα ονόματα των εν λόγω προγραμμάτων (pissb, piam, pifm, pisstv και pifsq), το καθένα χειρίζεται σήματα διαφορετικής διαμόρφωσης. Με μια πρώτη εξέταση, η αρθρωτή (modular) λογική που ακολουθεί το RPiTX μοιάζει άσκοπα περίπλοκη, αλλά όπως θα δείτε προσφέρει μεγάλη ευελιξία.



Το διάγραμμα των ακροδεκτών GPIO, σε ένα Raspberry Pi πρώτης γενιάς. Ο ακροδέκτης που αξιοποιεί το RPiTX, πάνω στον οποίο προσαρμόζουμε τη λιλιπούτεια κεραία, είναι ο υπ' αριθμόν 12 (GPIO 18).

Πριν εξετάσουμε τη χρήση του RPiTX και των εργαλείων που το συνοδεύουν, πρέπει να πούμε δυο λόγια για την εγκατάστασή τους. Ο πηγαίος κώδικας του προγράμματος βρίσκεται στο github και διανέμεται μαζί μ' ένα script, που αναλαμβάνει την εγκατάσταση των απαιτούμενων βιβλιοθηκών και τη μεταγλώττιση. Για να κατεβάσουμε τον πηγαίο κώδικα αρκεί να ανοίξουμε μια κονσόλα στο Raspberry Pi και να δώσουμε κάτι τέτοιο:

```
$ git clone https://github.com/F5OEO/rpitx.git
```

Η διαδικασία λήψης θα ολοκληρωθεί γρήγορα και θα δημιουργηθεί ένας κατάλογος με το όνομα rpitx. Το επόμενο βήμα είναι να εισέλθουμε στον κατάλογο και να εκτελέσουμε το script της εγκατάστασης:

```
$ cd rpitx
$ ./install.sh
```

Κατά τη διάρκεια της μεταγλώττισης θα ερωτηθούμε για τη λήψη και την εγκατάσταση διάφορων άλλων πακέτων. Για τη σωστή ολοκλήρωση της διαδικασίας πρέπει να απαντήσουμε καταφατικά σε όλες τις σχετικές ερωτήσεις. Μετά την εγκατάσταση, το rpitx κι όλα τα συνοδευτικά εργαλεία θα έχουν τοποθετηθεί στον κατάλογο /usr/bin και θα είναι έτοιμα προς χρήση.

Πριν αξιοποιήσουμε τα νέα εργαλεία πρέπει να κάνουμε μια μικρή "φυσική" προσθήκη στο Raspberry Pi. Αναφερόμαστε στην τοποθέτηση μιας κεραίας, που θα διευκολύνει τη μετάδοση του σήματος. Το μόνο που χρειαζόμαστε είναι ένα μικρό κομμάτι σύρματος (5cm είναι αρκετά), που θα πρέπει να προσαρμόσουμε στον ακροδέκτη GPIO 18. Παρεμπιπτόντως, οφείλουμε να επισημάνουμε ότι η αναμετάδοση ραδιοσυχνοτήτων απαιτεί ειδική άδεια. Στη δική μας περίπτωση, όμως, η εξαιρετικά περιορισμένη εμβέλεια του πομπού (βλέπε Raspberry Pi) είναι σίγουρο ότι δεν θα ενοχλήσει κανέναν.

Τα πρώτα βήματα

Για αρχή θα εξετάσουμε ένα απλό σενάριο, μόνο και μόνο για να δείτε τη γενική ιδέα. Θα μεταδώσουμε το περιεχόμενο ενός αρχείου ήχου, στα 100MHz και με διαμόρφωση FM. Όπως αντιλαμβάνεστε, κάτι τέτοιο θα μπορούσε να γίνει πανεύκολα και με το PiFM. Εμείς ωστόσο θα στηριχτούμε στο RPiTX και σε ένα από τα συνοδευτικά εργαλεία που ονομάζεται pifm. Αρχικά, με τη βοήθεια του pifm θα δημιουργήσουμε από το αρχείο ήχου ένα αρχείο FT και στη συνέχεια θα εκπέμψουμε το περιεχόμενο του εν λόγω αρχείου με τη βοήθεια του RPiTX.

```
$ pifm sampleaudio.wav fmaudio.ft
$ sudo rpitx -m RF -i fmaudio.ft -f 100000 -l
```

Η χρήση του pifm είναι απλή: Η πρώτη παράμετρος είναι το αρχείο ήχου που θέλουμε να μετατρέψουμε και η δεύτερη δηλώνει το όνομα του αρχείου που θα δημιουργηθεί. Περνώντας στο rpitx, παρατηρήστε ότι το εκτελούμε με τη βοήθεια του sudo. Το εν λόγω πρόγραμμα, βλέπετε, πρέπει να έχει πρόσβαση στους ακροδέκτες GPIO και αυτό προϋποθέτει δικαιώματα root. Η παράμετρος –m καθορίζει τον τρόπο λειτουργίας (mode) του προγράμματος και δεν υπάρχει λόγος να το ρίξουμε πάλι στη θεωρία. Στην περίπτωση της εκπομπής σημάτων FM επιλέγουμε τον τρόπο λειτουργίας RF. Με την παράμετρο –i ορίζουμε το όνομα του αρχείου που θέλουμε να μεταδώσουμε, ενώ με την παράμετρο –f δίνουμε τη συχνότητα εκπομπής σε KHz. Τέλος, η παράμετρος –l (από το loop) επιβάλλει την επαναλαμβανόμενη εκπομπή του αρχείου.

Ας δούμε ένα εξίσου απλό παράδειγμα. Ας υποθέσουμε ότι θέλουμε να εκπέμψουμε το περιεχόμενο ενός αρχείου ήχου, στη συχνότητα 1700kHz και με διαμόρφωση AM. Αυτή τη φορά, μαζί με το rpitx θα χρησιμοποιήσουμε και το piam:

```
$ piam sampleaudio.wav amaudio.ft
$ sudo rpitx -m RFA -i amaudio.ft -f 1700 -l
```

Νομίζουμε ότι δεν χρειάζεται κάποια ιδιαίτερη επεξήγηση. Το μόνο που αλλάζει είναι η χρήση του mode RFA και φυσικά, η συχνότητα εκπομπής που δίνουμε μετά την παράμετρο –f.

Εκπομπή εικόνας

Ας επιστρέψουμε τώρα στο θέμα μας. Η εικόνα πρέπει να έχει συγκεκριμένες διαστάσεις και βάθος χρώματος, σύμφωνα με το πρότυπο κωδικοποίησης που σκοπεύουμε να χρησιμοποιήσουμε. Το pisstv υποστηρίζει το πρότυπο Martin M1 κι αυτό σημαίνει ότι η εικόνα πρέπει να έχει διατάσεις 320x240 pixels και βάθος χρώματος 8 bits. Επιπρόσθετα, η εικόνα πρέπει να είναι αποθηκευμένη σε ασυμπίεστη μορφή αλλά, όπως καταλαβαίνετε, αυτό δεν αποτελεί περιορισμό του προτύπου. Πρόκειται για μια ιδιαιτερότητα του τρόπου λειτουργίας του pisstv. Τελικά, για να εκπέμψουμε μια εικόνα πρέπει να κάνουμε τα εξής: Αρχικά πρέπει να εξασφαλίσουμε ότι έχει τις σωστές διαστάσεις και το αναμενόμενο βάθος χρώματος. Κάτι τέτοιο μπορεί να γίνει ακόμα και με το πιο απλό σχεδιαστικό πρόγραμμα και δεν χρειάζεται κάποια περαιτέρω εξήγηση. Στη συνέχεια πρέπει να μετατρέψουμε το αρχείο εικόνας σε μια μορφή κατανοητή από το pisstv. Ο πιο απλός και γρήγορος τρόπος είναι με τη βοήθεια του πακέτου ImageMagick. Αν δεν το έχετε εγκαταστήσει ήδη στο Raspberry, αρκεί να δώσετε κάτι τέτοιο:

```
$ sudo apt-get install imagemagick
```

Ακολούθως, για τη μετατροπή της εικόνας σε ασυμπίεστη μορφή μπορούμε να χρησιμοποιήσουμε το πρόγραμμα convert:

```
$ convert -depth 8 picture.jpg picture.rgb
```

Τώρα μπορούμε να τρέξουμε το pisstv, για να δημιουργήσουμε το αρχείο FT που στη συνέχεια θα εκπέμψει το RPiTX:

```
$ pisstv picture.rgb picture.ft
$ sudo rpitx -m RF -i picture.ft -f 14230
```

Αυτό ήταν όλο. Η πρώτη μας μετάδοση εικόνας SSTV είναι γεγονός!

Μετά από αυτό το μικρό κατόρθωμα ανοίγονται μπροστά μας απεριόριστες δυνατότητες. Τι θα λέγατε αν εκπέμπαμε την εικόνα που λαμβάνουμε από μια μικρή κάμερα; Ο καλύτερος υποψήφιος γι' αυτή τη δουλειά είναι το Raspberry Pi Camera Module. Για να πετύχουμε το σκοπό μας, κατασκευάσαμε ένα απλούστατο BASH script:

```
#!/bin/bash
echo "Press [CTRL+C] at any moment to stop..."
while :
do
      raspistill -w 320 -h 256 -o ~/img.jpg -t 1
      convert -depth 8 ~/img.jpg ~/img.rgb
      pisstv ~/img.rgb ~/img.ft
      sudo rpitx -m RF -i ~/img.ft -f 14230
done
```

Όπως βλέπετε, το σκριπτάκι τυπώνει ένα μήνυμα και στη συνέχεια εισέρχεται σε έναν ατέρμονα βρόχο. Εκεί μέσα γίνεται όλη η δουλειά. Κατ' αρχάς, με τη βοήθεια του raspistill παίρνουμε ένα στιγμιότυπο (still) από το οπτικό πεδίο της κάμερας. Στη συνέχεια, με τον τρόπο που είδαμε λίγο νωρίτερα εκπέμπουμε την εικόνα και η διαδικασία επαναλαμβάνεται.

Sound-check για την εικόνα

Για τους φίλους ραδιοερασιτέχνες, ο περιορισμός στους τέσσερεις τοίχους τους σπιτιού είναι ανυπόφορος. Η χρήση του Raspberry Pi σε συνδυασμό με το RPiTX δεν αποτελεί "σοβαρή" λύση. Ένας ραδιοερασιτέχνης θα προτιμούσε να κωδικοποιήσει την εικόνα σύμφωνα με κάποιο πρότυπο του SSTV και στη συνέχεια να λάβει το αποτέλεσμα με τη μορφή ενός αρχείου ήχου. Έτσι, θα μπορούσε να εκπέμψει τον ήχο (δηλαδή την εικόνα) μέσω του πανίσχυρου ραδιοερασιτεχνικού εξοπλισμού. Ε,

λοιπόν, αν κι αυτή η διαδικασία μοιάζει αδιάφορη για τους περισσότερους από εμάς, πετυχαίνει έναν uber-geeky σκοπό: Θα μας επιτρέψει να ακούσουμε μια εκπομπή SSTV!

Αν νομίζετε ότι σκοπεύουμε να μπλέξουμε με περισσότερα εξωτικά εργαλεία, σφάλλετε τα μάλα. Η βιβλιοθήκη PySSTV μπορεί να πετύχει όλα όσα θέλουμε, χωρίς πολλά-πολλά. Όπως αντιλαμβάνεστε, πρόκειται για βιβλιοθήκη της αγαπημένης μας γλώσσας προγραμματισμού, της Python. Με τη βοήθειά της, καθώς και τη συνδρομή της PIL (Python Image Library), μπορούμε να χρησιμοποιούμε όλα τα δημοφιλή φορμά εικόνας (BMP, PNG, JPEG, GIF κ.ά.) και να παράγουμε σήμα SSTV στη μορφή αρχείων WAV. Η PySSTV υποστηρίζει μια ικανοποιητική γκάμα από πρότυπα κωδικοποίησης, όπως τα Martin M1, Martin M2, Robot B/W 8, Robot B/W 24, Scottie S1, Pasokon P3 κ.ά. Ο κώδικας της βιβλιοθήκης βρίσκεται στο αποθετήριο https://pypi.python.org/pypi/PySSTV και η εγκατάστασή της πραγματοποιείται πανεύκολα, είτε με τη βοήθεια των setup tools είτε με τη χρήση του εξαιρετικού package manager ονόματι pip. Αυτή τη φορά δεν χρειάζεται να δουλέψουμε στο Raspberry Pi. Μπορούμε να στραφούμε σε κάποιο "μεγάλο" σύστημα που τρέχει Linux ή και σε κάποια εικονική μηχανή. Στο δικό μας σύστημα είχαμε ήδη εγκατεστημένο το pip και το μόνο που χρειάστηκε να κάνουμε ήταν να ανοίξουμε μια κονσόλα:

```
$ pip install pysstv
```

Με την ευκαιρία, σημειώστε ότι μπορείτε να μάθετε τα πάντα για τους package manager της Python στο άρθρο "Βγάλτε βόλτα την Python" του τεύχους 034 (http://deltahacker.gr/deltahacker034). Εκτελώντας το παραπάνω, μέσα σε λίγα δευτερόλεπτα η βιβλιοθήκη PySSTV θα έχει εγκατασταθεί και θα είναι έτοιμη προς χρήση. Τώρα, το μόνο που μένει είναι να φτιάξουμε ένα πρόγραμμα που θα την αξιοποιεί. Σωστά; Και ναι, και όχι. Ο "φυσιολογικός" τρόπος χρήσης είναι ακριβώς αυτός που περιγράψαμε, αλλά εμείς θα καταφύγουμε σε μια πρόχειρη και γρήγορη λύση. Θα χρησιμοποιήσουμε τη βιβλιοθήκη σαν ένα module της Python, στο οποίο θα περάσουμε τις επιθυμητές παραμέτρους μονομιάς:

```
$ python -m pysstv --mode MartinM1 input.png output.wav
```

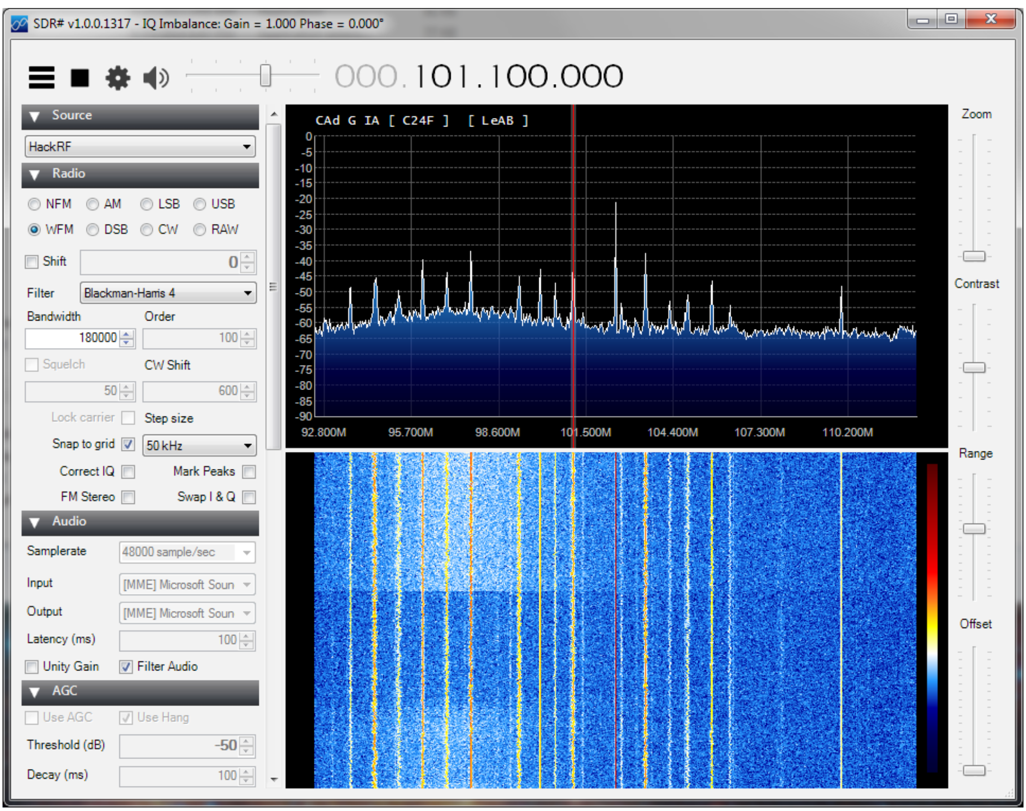
Προφανώς, το input.png αποτελεί την εικόνα που θέλουμε να εκπέμψουμε και το output.wav αποτελεί το αρχείο ήχου με το σήμα SSTV. Τώρα, μπορούμε να ανοίξουμε έναν audio player και να ακούσουμε ...την εικόνα. Δεν ξέρουμε τι θα ακούσετε εσείς και πώς θα σας φανεί, αλλά τα δικά μας αφτιά βουίζουν ακόμα.

Ακούει κανείς;

Για να οπτικοποιήσει κανείς τις εκπομπές μας, για να καταφέρει να δει τις εικόνες που εκπέμπουμε, χρειάζονται ειδικά προγράμματα. Για καλή μας τύχη υπάρχουν αρκετά δωρεάν εργαλεία. Εμείς δοκιμάσαμε το MMSSTV και το RX-SSTV, αλλά αυτό

που μας κέρδισε ήταν το πρώτο. Μπορείτε να το κατεβάσετε από τη διεύθυνση http://hamsoft.ca/pages/mmsstv.php. Για τις πρώτες δοκιμές καταφύγαμε σε μια πονηριά. Αντί να εκπέμψουμε το σήμα SSTV με ραδιοκύματα, το στείλαμε στα ηχεία μας. Ταυτόχρονα, ρυθμίσαμε το MMSSTV για να χρησιμοποιήσει σαν είσοδο το μικρόφωνο. Αν και το μικρόφωνο "έπιανε" όλους τους θορύβους του περιβάλλοντος, το αποτέλεσμα ήταν ικανοποιητικότατο. Έτσι, εφόσον είχαμε πειστεί ότι οι παράξενοι ήχοι αντιστοιχούσαν πράγματι σε μια εκπομπή SSTV, προχωρήσαμε σε μια πιο ρεαλιστική δοκιμή.

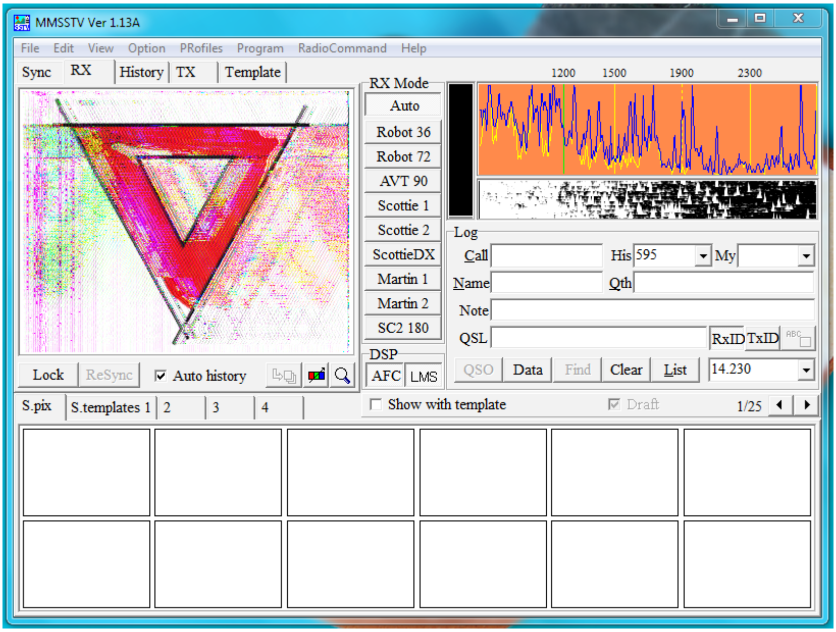
Για την ηλεκτρομαγνητική εκπομπή του σήματος χρησιμοποιήσαμε το Raspberry Pi, το RPiTX κι ένα μικρό καλώδιο σε ρόλο κεραίας. Κατά την εκτέλεση του RPiTX επιλέξαμε τη συχνότητα των 100MHz και δώσαμε την παράμετρο –l, για να πετύχουμε τη διαρκή εκπομπή της εικόνας. Έχοντας ξεμπερδέψει πλέον με τον πομπό, στραφήκαμε στο δέκτη. Για τη λήψη αξιοποιήσαμε την τεχνική Software-Defined Radio κι έναν μικρό δέκτη DVB από την Terratec. Αφού συνδέσαμε το δέκτη στο σύστημα, το πρώτο πράγμα που κάναμε ήταν να αντικαταστήσουμε τον



Το περιβάλλον εργασίας του SDR# και ειδικά το πάνελ ρυθμίσεων που εμφανίζεται στ' αριστερά, μοιάζει πολύ περίπλοκο. Αν το μελετήσετε με ψυχραιμία, πάντως, θα διαπιστώσετε ότι όλες οι ρυθμίσεις είναι απλές και κατανοητές. Το υπόλοιπο περιβάλλον θα φανεί αδιάφορο σε όσους δεν έχουν εντρυφήσει στο θέμα, αλλά είναι σίγουρα εντυπωσιακό :D

"επίσημο" driver που εγκαταστάθηκε αυτόματα. Για το σκοπό αυτό κατεβάσαμε και εγκαταστήσαμε το πρόγραμμα Zadig (http://zadig.akeo.ie). Αμέσως μετά, κατεβάσαμε κι εγκαταστήσαμε το πρόγραμμα SDR# (http://sdrsharp.com). Από το περιβάλλον εργασίας του Zadig επιλέξαμε τον τηλεοπτικό μας δέκτη, στην περίπτωσή μας ήταν το "Bulk-In, Interface (Interface 0)", και πατήσαμε το κουμπί "Replace Driver". Όταν ολοκληρώθηκε η διαδικασία αντικατάστασης τρέξαμε το SDR# και ως συσκευή εισόδου επιλέξαμε το "USB-RLT". Έτσι, ο δέκτης τέθηκε σε λειτουργία κι εμείς ρυθμίσαμε το SDR# για λήψη στη συχνότητα του πομπού μας (στα 100MHz). Αμέσως μόλις συντονιστήκαμε άρχισαν να ακούγονται τα "κελαϊδίσματα" του Raspberry Pi. Σε αυτή τη φάση ανοίξαμε το MMSSTV κι επιλέξαμε σαν είσοδο το μικρόφωνο. Για μία ακόμα φορά, τα είχαμε καταφέρει! Βέβαια, όπως και στην πρώτη δοκιμή, οι θόρυβοι του δωματίου αλλοίωναν το σήμα που λάμβανε το μικρόφωνο και κατ' επέκταση αλλοίωναν το περιεχόμενο της εκπομπής μας. Αυτό το πρόβλημα θα μπορούσε να λυθεί οριστικά με τη βοήθεια άλλων προγραμμάτων, που θα τροφοδοτούσαν την έξοδο του SDR στην είσοδο του αποκωδικοποιητή, χωρίς τη μεσολάβηση των ηχείων και του μικροφώνου. Εμείς κάναμε κάτι πολύ πιο απλό: Πήραμε ένα καλώδιο mini-jack σε mini-jack και "γεφυρώσαμε" το line-out με το line-in. Στη συνέχεια, από το περιβάλλον του MMSSTV επιλέξαμε σαν είσοδο το line-in. Με αυτό το κολπάκι, το ηχητικό σήμα που λάμβανε το SDR# (το σήμα SSTV) έφτανε στον αποκωδικοποιητή αναλλοίωτο ;-)

Παίζοντας με το Raspberry Pi και τις εκπομπές SSTV, αρχίσαμε να σκεφτόμαστε διάφορα παράξενα projects. Ποια τρελή ιδέα γυροφέρνει άραγε στη σκέψη σας; Όποια κι αν είναι, μη διστάσετε να τη μοιραστείτε μαζί μας και, πού ξέρετε, ίσως στο μέλλον να την υλοποιήσουμε.



Μετά από αρκετές δοκιμές και μικρορυθμίσεις στο SDR#, το MMSSTV κατάφερε να εμφανίσει την εικόνα που εξέπεμπε το Raspberry Pi: το λογότυπο του αγαπημένου μας περιοδικού.

Skill: Intermediate
Tags: Symmetric key cryptography, public key cryptography, public keys, private keys, certificates, Web-of-Trust, Certification Authorities, Keybase, proofs, PGP, GnuPG, OpenPGP

Κρυπτογραφία δημοσίου κλειδιού:

Το πρόβλημα της εμπιστοσύνης και μια έξυπνη λύση

Στα τόσα καλά που μας προσφέρουν τα μαθηματικά της κρυπτογραφίας δημοσίου κλειδιού, υπάρχει και κάτι που πολύ θα το θέλαμε αλλά δεν είναι εύκολο να το έχουμε. Πώς μπορούμε να είμαστε βέβαιοι για την πραγματική ταυτότητα των κατόχων δημοσίων κλειδιών; Λύσεις υπάρχουν, καμία όμως δεν είναι τέλεια. Τον τελευταίο καιρό πάντως δοκιμάζεται μία νέα, έξυπνη και πολλά υποσχόμενη προσέγγιση στο όλο θέμα.

του Χρήστου Βαρελά

Στο παρόν άρθρο σκοπεύουμε να κεντρίσουμε το ενδιαφέρον σας περί της κρυπτογραφίας δημοσίου κλειδιού. Ακόμη κι αν έχετε ήδη ασχοληθεί αλλά βρίσκετε ότι η εφαρμογή των τεχνικών κρυπτογράφησης κι επαλήθευσης ταυτότητας είναι αρκετά φασαριόζικες για ν' αξίζουν τον κόπο, πιστεύουμε ότι έχουμε καλές πιθανότητες ώστε ν' αναθερμάνουμε το ενδιαφέρον σας.

Θα ξεκινήσουμε με μια γρήγορη εισαγωγή στην κρυπτογραφία γενικότερα και στην κρυπτογραφία δημοσίου κλειδιού ειδικότερα. Αμέσως μετά θα συζητήσουμε για ένα σημαντικό πρόβλημα, το οποίο αφορά στην επαλήθευση της πραγματικής ταυτότητας των κατόχων δημοσίων κλειδιών. Σκεφτείτε το: Δεν έχει και πολύ αξία να στέλνουμε ισχυρά κρυπτογραφημένα μηνύματα σε κάποιον, όταν αυτός ο κάποιος *δεν* είναι εκείνος που νομίζουμε. Μια πολλά υποσχόμενη λύση στο συγκεκριμένο πρόβλημα αποτελεί το Keybase: Είναι μια νέα, πολλά υποσχόμενη υπηρεσία, η οποία επιτίθεται στο πρόβλημα της ταυτοποίησης με τις λεγόμενες αποδείξεις (proofs). Πολλά περισσότερα θα πούμε σε λίγο, αλλά προς το παρόν ας ξεκινήσουμε βάζοντας τα πράγματα σε μία σειρά.

Τι είναι η κρυπτογραφία;

Πρόκειται για τη μελέτη τεχνικών απόκρυψης του περιεχομένου μηνυμάτων, τα οποία διακινούνται με όχι κατ' ανάγκη ασφαλείς τρόπους ή μέσω επισφαλών καναλιών επικοινωνίας. Το περιεχόμενο των κρυπτογραφημένων μηνυμάτων πρέπει να είναι διαθέσιμο μόνο στους επιθυμητούς παραλήπτες. Μόνον εκείνοι, δηλαδή, πρέπει να 'ναι σε θέση να τα αποκρυπτογραφούν. Σημειώστε εδώ ότι η υποκλοπή των κρυπτογραφημένων μηνυμάτων είναι γενικά εφικτή – και συχνά εύκολη υπόθεση. Όταν όμως η μέθοδος της κρυπτογράφησης δουλεύει, μ' άλλα λόγια όταν είναι ισχυρή, τότε οι υποκλοπείς αδυνατούν ν' αποκαλύπτουν το περιεχόμενο των μηνυμάτων.

Ιστορικά έχουν χρησιμοποιηθεί αρκετές μέθοδοι για την κρυπτογράφηση μηνυμάτων. Για παράδειγμα, σε κάθε εισαγωγικό κείμενο είναι μάλλον απίθανο να μη διαβάσετε για τη μέθοδο της αντικατάστασης του Καίσαρα (**Ceasar's Cipher**), κατά την οποία κάθε γράμμα του αρχικού κειμένου αντικαθίσταται από ένα γράμμα που βρίσκεται σταθερό αριθμό θέσεων πιο κάτω στο αλφάβητο. (Όταν φτάνουμε προς το τέλος του αλφαβήτου η αντικατάσταση "υπερχειλίζει" προς την αρχή, κυκλικά.)

Στο παρόν άρθρο δεν θα κάνουμε ιστορική αναδρομή στις τεχνικές κρυπτογράφησης. Όποιος βέβαια ενδιαφέρεται να μάθει περισσότερα, δεν θα δυσκολευτεί να βρει απειραριθμήσιμο πλήθος εισαγωγικών άρθρων στο web. Για τη συνέχεια, ας σημειώσουμε μόνο ότι το αρχικό κείμενο, πριν την κρυπτογράφησή του, ονομάζεται **plaintext**, ενώ η κρυπτογραφημένη του εκδοχή, δηλαδή το αρχικό κείμενο *μετά* την κρυπτογράφησή του, ονομάζεται **ciphertext**.

Συμμετρική κρυπτογραφία

Από την προηγούμενη ενότητα φαίνεται ότι η κρυπτογραφία αφορά στη μετατροπή του plaintext σε ciphertext. Ειδικά από το παράδειγμα του Ceasar's Cipher, καταλαβαίνουμε ότι η επαναφορά του plaintext από το ciphertext γίνεται με συγκεκριμένο, γνωστό τρόπο. Αλλά τον τρόπο αυτό δεν μπορεί κάποιος να τον εφαρμόσει – εκτός αν γνωρίζει ένα καλά κρυμμένο μυστικό. Είναι σημαντικό να υπογραμμίσουμε ότι, ειδικά στο πλαίσιο της σύγχρονης κρυπτογραφίας, ο τρόπος κατά τον οποίο το plaintext γίνεται ciphertext *δεν* είναι μυστικός. Ο αλγόριθμος της κρυπτογράφησης, μ' άλλα λόγια, είναι γνωστός σε όλους. Απλά, προκειμένου να εφαρμόσει κάποιος τον αλγόριθμο αντίστροφα, ώστε να πάρει το plaintext από το ciphertext, πρέπει να διαθέτει κι ένα κατάλληλο κλειδί (**key**): αυτό είναι το καλά κρυμμένο μυστικό.

Στο παράδειγμα του Ceasar's Cipher, ο αλγόριθμος της κρυπτογράφησης είναι η "ολίσθηση" κατά ένα σταθερό πλήθος θέσεων πιο κάτω στο αλφάβητο. Τον αλγόριθμο τον γνωρίζουν όλοι: αποστολείς, παραλήπτες, κακόβουλοι χρήστες, αμέριμνοι διαβάτες, καστανάδες, φωνακλάδες τουρίστες, οι ιθαγενείς στην Παπούα Νέα Γουινέα, η κεντρική επιτροπή του Σύριζα – όλοι, μα όοοοολοι ανεξαιρέτως. Αυτό όμως που δεν γνωρίζουν όλοι, παρά μόνον ο αποστολέας και οι επιθυμητοί παραλήπτες, είναι το κλειδί που χρησιμοποιείται ανά περίπτωση. Στο απλό παράδειγμά μας, το κλειδί δεν είναι παρά το *ακριβές* πλήθος των θέσεων μετατόπισης στο αλφάβητο. Από τη στιγμή που ο παραλήπτης γνωρίζει τον αλγόριθμο κρυπτογράφησης αλλά

και το κλειδί, είναι εύκολο να εφαρμόσει τον αντίστροφο αλγόριθμο πάνω στο ciphertext –αντί να πηγαίνει k θέσεις μπροστά θα πηγαίνει k θέσεις πίσω– και πολύ εύκολα θα αποκτήσει το plaintext.

Το κλειδί, λοιπόν, πρέπει πάντοτε να προστατεύεται και να το γνωρίζουν μόνον όσοι χρειάζεται να το γνωρίζουν. Μάλιστα για λόγους ασφαλείας, από καιρού εις καιρόν το κλειδί καλό είναι ν' αλλάζει, οπότε και οι παραλήπτες θα πρέπει, με κάποιον τρόπο, να ενημερώνονται σχετικά. Πάντως η προστασία του κλειδιού δεν είναι πανάκεια. Ειδικά σήμερα που η υπολογιστική ισχύς περισσεύει και είναι διαθέσιμη σε όλους, ορισμένοι αλγόριθμοι κρυπτογράφησης είναι παντελώς αδύναμοι – κι ας μη διαρρέουν ποτέ τα κλειδιά κρυπτογράφησης. Ας δούμε ξανά το παράδειγμα του Ceasar's Cipher. Γνωρίζοντας κάποιος ότι έχει χρησιμοποιηθεί για την κρυπτογράφηση ενός μηνύματος, με μια επίθεση brute force είναι πολύ εύκολο να πάρει το αρχικό μήνυμα. Στη συγκεκριμένη περίπτωση, μάλιστα, επιθέσεις πολύ πιο αποδοτικές από τις "χοντροκομμένες" brute force είναι εφικτές. Ακόμη κι αν δεν έχουμε καν γνώση του αλγορίθμου κρυπτογράφησης, εφαρμόζοντας τεχνικές στατιστικής ανάλυσης που βασίζονται στη συχνότητα εμφάνισης των γραμμάτων σε κανονικά (μη-κρυπτογραφημένα) κείμενα μιας γλώσσας, εύκολα ξεκινάμε από το ciphertext και καταλήγουμε στο plaintext.

Οι σύγχρονοι, ισχυροί αλγόριθμοι κρυπτογράφησης, παραμένουν γνωστοί αλλά *δεν* είναι ευάλωτοι σε επιθέσεις όπως, π.χ., εκείνη της στατιστικής ανάλυσης. Μόνη ελπίδα για έναν επιτιθέμενο είναι να βρει το κλειδί της κρυπτογράφησης. Όμως τα κλειδιά έχουν το κατάλληλο μήκος (σε bits) και η όποια εξαντλητική επίθεση brute force απαιτεί τόσο πολύ χρόνο, ώστε στην πράξη να είναι παντελώς άχρηστη.

Στην έως τώρα συζήτησή μας, θα παρατηρήσατε ότι το κλειδί της κρυπτογράφησης είναι ίδιο με εκείνο της αποκρυπτογράφησης. Ακριβώς για το λόγο αυτό, μιλάμε για κρυπτογραφία συμμετρικού κλειδιού (**symmetric-key cryptography**). Ασχέτως της "ποιότητας" του αλγορίθμου κρυπτογράφησης ή του μήκους του κλειδιού, η συμμετρική κρυπτογραφία έχει ένα σοβαρό πρόβλημα: με κάποιον τρόπο, οι παραλήπτες των κρυπτογραφημένων μηνυμάτων πρέπει να γνωρίζουν το κλειδί. Πώς επιτυγχάνεται κάτι τέτοιο; Μία συνάντηση είναι ένας καλός κι ασφαλής τρόπος, όμως δεν είναι πάντοτε εφικτός. Είναι και το άλλο: Αν πρόκειται να συναντήσουμε κάποιον για να του δώσουμε το κλειδί για ένα κρυπτογραφημένο μήνυμα που θέλουμε να του στείλουμε, ε, μπορούμε τότε να του μεταφέρουμε και το ίδιο το μήνυμα. Για να έχει λοιπόν η κρυπτογραφία συμμετρικού κλειδιού πρακτική αξία, τα εμπλεκόμενα κλειδιά πρέπει να αποστέλλονται στους παραλήπτες, π.χ., μέσω Internet. Φυσικά, η αποστολή πρέπει να γίνεται με τέτοιον τρόπο ώστε η υποκλοπή να είναι αδύνατη. Στο σημείο αυτό θα μπορούσε κάποιος να προτείνει ότι η αποστολή θα πρέπει να λαμβάνει χώρα πάντοτε μέσω ενός κρυπτογραφημένου καναλιού, όμως τότε το πρόβλημα απλά μετατίθεται λίγο πιο κάτω: πώς προστατεύουμε το κλειδί για την κρυπτογράφηση του ίδιου του καναλιού μετάδοσης;

Κρυπτογραφία δημοσίου κλειδιού

Το πρόβλημα της ασφαλούς μετάδοσης του κλειδιού έρχεται να αντιμετωπίσει η κρυπτογραφία δημοσίου κλειδιού (**public key cryptography**), η οποία βασίζεται σε μαθηματικά προβλήματα τα οποία δεν επιδέχονται αποδοτικές λύσεις. Παράδειγμα: Για το πρόβλημα της παραγοντοποίησης των ακεραίων αριθμών ή για εκείνο του υπολογισμού του διακριτού λογαρίθμου σε πεπερασμένες ομάδες, μέχρι σήμερα δεν έχει βρεθεί λύση που να ολοκληρώνεται σε πρακτικό για τα ανθρώπινα δεδομένα χρόνο – και δεν έχει καν σημασία η ισχύς των υπολογιστών που θα μπορούσαν να επιστρατευτούν για την εύρεση μιας λύσης.

Αναλυτικότερα, στο πλαίσιο της κρυπτογραφίας δημοσίου κλειδιού αντί για ένα έχουμε δύο, μαθηματικά συσχετιζόμενα, κλειδιά. Το ένα εξ αυτών ονομάζεται ιδιωτικό (**private key**) και το άλλο δημόσιο (**public key**). Το ιδιωτικό κλειδί φυλάσσεται ως κόρη οφθαλμού, ενώ το δημόσιο διανέμεται ελεύθερα σε κάθε ενδιαφερόμενο. Προσέξτε εδώ ότι αν και τα δύο κλειδιά συσχετίζονται μαθηματικά, ο υπολογισμός του ιδιωτικού ξεκινώντας από το δημόσιο είναι πρακτικά αδύνατος.

Σε κάθε ζεύγος ιδιωτικού/δημοσίου κλειδιού, το δημόσιο το χρησιμοποιούν όσοι θέλουν να μας στέλνουν κρυπτογραφημένα μηνύματα. Αυτά, αποκρυπτογραφούνται από το αντίστοιχο ιδιωτικό κλειδί και μόνον από αυτό. Από τη στιγμή που μόνον εμείς έχουμε το ιδιωτικό κλειδί, ο αποστολέας του κρυπτογραφημένου μηνύματος δικαιούται να είναι ήσυχος με τη γνώση ότι μόνον εμείς θα μπορέσουμε να το αποκρυπτογραφήσουμε. Προσέξτε: Ένα σωρό χρήστες είναι πιθανό να αποκτήσουν το κρυπτογραφημένο μήνυμα. Ακόμη κι αν γνωρίζουν καλά τον αλγόριθμο που χρησιμοποιήθηκε για την παραγωγή του ciphertext, από τη στιγμή που *δεν* διαθέτουν το αντίστοιχο ιδιωτικό κλειδί δεν μπορούν να κάνουν κάτι.

Μια άλλη χρήση του ιδιωτικού κλειδιού εκτός της αποκρυπτογράφησης, είναι για την παραγωγή των λεγόμενων ψηφιακών υπογραφών (**digital signatures**). Πριν στείλουμε ή δημοσιεύσουμε ένα μήνυμα, παράγουμε μια σύνοψή του (**message digest**) την οποία και κρυπτογραφούμε με χρήση του ιδιωτικού μας κλειδιού. Με άλλα λόγια, αυτό που κάνουμε είναι να υπογράψουμε ψηφιακά το μήνυμα. Γιατί μιλάμε για ψηφιακή υπογραφή; Πολύ απλά, διότι η κρυπτογραφημένη σύνοψη αποκρυπτογραφείται *μόνο* με χρήση του αντίστοιχου δημόσιου κλειδιού. Έτσι, όποιος ενδιαφέρεται για το γνήσιο της υπογραφής, απλά επιχειρεί να αποκρυπτογραφήσει τη σύνοψη με το δημόσιο κλειδί μας. Αν τα καταφέρει –κι επειδή υποτίθεται ότι μόνον εμείς κατέχουμε το αντίστοιχο ιδιωτικό κλειδί–, δικαιούται να είναι σίγουρος για το γνήσιο της υπογραφής. Επιπρόσθετα, αν η σύνοψη αποτελεί το αποτέλεσμα μιας συνάρτησης κατακερματισμού (**hash function**) επί του αρχικού κειμένου, τότε εκτός από το γνήσιο της υπογραφής κάθε ενδιαφερόμενος επαληθεύει και την ακεραιότητα (**integrity**) του μηνύματος που μόλις έλαβε. Βεβαιώνεται, δηλαδή, ότι το μήνυμα δεν τροποποιήθηκε κατά την αποστολή του.

Για το πώς δουλεύει η κρυπτογραφία δημοσίου κλειδιού στην πράξη και, ειδικότερα, για το πώς εκμεταλλευόμαστε τις σχετικές τεχνικές προκειμένου να στέλνουμε και να λαμβάνουμε κρυπτογραφημένη ή/και ψηφιακά υπογεγραμμένη ηλεκτρονική αλληλογραφία, παρακολουθήστε τα δύο εκτενή video tutorials στο http://deltahacker.gr/?p=14213.

Ζήτημα εμπιστοσύνης

Κάθε φορά που αποφασίζουμε ότι θέλουμε να έχουμε την επιλογή για αποστολή κρυπτογραφημένων email ή/και αρχείων σε κάποιον, τότε το πρώτο πράγμα που χρειαζόμαστε είναι το λεγόμενο δημόσιο πιστοποιητικό (**public certificate**) του "κάποιου". Πρόκειται για ένα αρχείο απλού κειμένου με συγκεκριμένη δομή, το οποίο μεταξύ άλλων περιλαμβάνει μία ή περισσότερες διευθύνσεις email και ονόματα, ένα δημόσιο κλειδί, μία ή περισσότερες ψηφιακές υπογραφές από άλλους χρήστες κ.ά. Για παράδειγμα, το δημόσιο πιστοποιητικό του γράφοντα είναι στο https://keybase.io/christos_varelas/key.asc (περισσότερα για το Keybase σε λίγο). Όποιος θέλει να του στέλνει κρυπτογραφημένα email ή/και αρχεία, ελεύθερα μπορεί να πάρει το εν λόγω πιστοποιητικό και να το εισάγει στο δημόσιο keyring του. Με χρήση του ίδιου πιστοποιητικού, εκτός από την αποστολή κρυπτογραφημένων μηνυμάτων ή/και αρχείων θα μπορεί να επαληθεύει και τις ψηφιακές υπογραφές του γράφοντα.

Το θέμα με τα δημόσια πιστοποιητικά δεν είναι η ποσότητα των πληροφοριών που περιέχουν. Αντίθετα, το πρόβλημα είναι ότι κανείς δεν είναι σε θέση να μας διασφαλίσει ότι ο κάτοχος ενός πιστοποιητικού είναι πράγματι αυτός που νομίζουμε ότι είναι. Ακούγεται λίγο παράξενα αυτή η πρόταση; Είναι παράξενη! Αν όμως το σκεφτούμε λίγο, θα συνειδητοποιήσουμε ότι ο ισχυρισμός για την κατοχή ενός πιστοποιητικού είναι ένα πράγμα, η απόδειξη όμως για την κατοχή ενός πιστοποιητικού από ένα συγκεκριμένο φυσικό πρόσωπο είναι εντελώς διαφορετικό πράγμα.

Ας πούμε, π.χ., ότι λαμβάνετε το δικό μου δημόσιο πιστοποιητικό. Είδατε τα δύο video tutorials στο http://deltahacker.gr/?p=14213, πωρωθήκατε με την κρυπτογραφία δημοσίου κλειδιού, φτιάξατε το δικό σας ζεύγος ιδιωτικού/δημοσίου κλειδιού κι αυτή τη στιγμή σκέφτεστε ότι δεν βλάπτει αν εισάγετε στο δημόσιο keyring σας το πιστοποιητικό μου. Εξετάζοντάς το, θα δείτε ότι περιλαμβάνει μια πληθώρα διευθύνσεων email. Κάποιες εξ αυτών, όπως εκείνες στο defiant.gr και στο colder.xyz, είναι λίγο παράξενες και δεν σας λένε πολλά. Κάποιες άλλες, όπως εκείνη στο deltahacker.gr, κάτι σας λένε. Αλλά υπάρχει ένα πρόβλημα. Όποιος θέλει, πολύ εύκολα μπορεί να δημιουργήσει ένα δημόσιο πιστοποιητικό που θα περιλαμβάνει ένα σωρό διευθύνσεις email – μεταξύ αυτών κι εκείνες που εμπεριέχονται στο δικό μου πιστοποιητικό. Η ελευθερία συμπερίληψης διευθύνσεων email στα πιστοποιητικά ισχύει *και* για τα ονόματα. Ξαφνικά, λοιπόν, είναι πολύ πιθανό να υπάρχουν περισσότερα από ένα δημόσια πιστοποιητικά στα email των οποίων περιλαμβάνονται κοινές διευθύνσεις, ενώ και το όνομα του κατόχου σε κάθε πιστοποιητικό είναι "Χρήστος Βαρελάς". Πώς στην ευχή, λοιπόν, βεβαιώνεται κάποιος για τον πραγματικό κάτοχο του πιστοποιητικού που τώρα εξετάζει; Ένας τρόπος είναι να συναντήσει τον υποτιθέμενο κάτοχο ή έστω να του τηλεφωνήσει. Κάπως έτσι, θα μπορέ-

σει να βγάλει ασφαλή συμπεράσματα. Καταλαβαίνετε φυσικά ότι, πλην ελαχίστων εξαιρέσεων, η λύση της συνάντησης ή του τηλεφώνου δεν είναι πρακτική. Για την ακρίβεια, στη συντριπτική πλειονότητα των περιπτώσεων δεν υφίσταται καν ως επιλογή. Το πρόβλημα, λοιπόν, παραμένει: Πώς διασφαλίζουμε ότι ένα συγκεκριμένο φυσικό πρόσωπο είναι πράγματι ο ιδιοκτήτης του δημόσιου πιστοποιητικού που τώρα εξετάζουμε και σπαζοκεφαλιάζουμε;

Υποδομές δημοσίων κλειδιών και ιστός της εμπιστοσύνης

Υπάρχουν διάφοροι τρόποι για την αντιμετώπιση του προβλήματος της εμπιστοσύνης, εννοείται στο πλαίσιο συστημάτων όπου γίνεται χρήση τεχνικών κρυπτογραφίας δημοσίου κλειδιού. Ένας εξ αυτών είναι με την εγκαθίδρυση μιας Υποδομής Δημοσίου Κλειδιού (**Public Key Infrastructure** ή απλά **PKI**). Κάθε PKI εμπεριέχει ένα σύνολο πολιτικών και διαδικασιών, οι οποίες ακολουθούνται για τη δημιουργία, την αποθήκευση, τη διαχείριση, τη χρήση αλλά και την ακύρωση δημόσιων πιστοποιητικών. Οι εν λόγω πολιτικές και διαδικασίες υλοποιούνται από το κατάλληλο hardware και software, αλλά και με τη συμμετοχή διαχειριστών και απλών χρηστών. Βασικός λόγος ύπαρξης μιας PKI είναι η αντιστοίχιση δημοσίων πιστοποιητικών σε οντότητες (**entities**). Η συγκεκριμένη, κρίσιμη εργασία, είναι ευθύνη μιας Αρχής Πιστοποίησης (**Certification Authority** ή απλά **CA**). Στο πλαίσιο λοιπόν μιας συγκεκριμένης PKI, εμπιστευόμαστε την CA και δεχόμαστε ότι έχει κάνει όλους τους απαραίτητους ελέγχους ώστε να μην υπάρχουν αμφιβολίες για τις αντιστοιχίσεις πιστοποιητικών σε οντότητες. (Ο λόγος που αντί να μιλάμε για χρήστες προτιμάμε το γενικότερο όρο "οντότητα", είναι διότι τα δημόσια πιστοποιητικά δεν αντιστοιχίζονται μόνο σε φυσικά πρόσωπα αλλά και σε δικτυακούς τόπους ή σε δικτυακές υπηρεσίες γενικότερα.)

Οι Αρχές Πιστοποίησης χρησιμοποιούνται κατά κόρον για την έκδοση δημοσίων πιστοποιητικών για web sites ή για άλλες δικτυακές υπηρεσίες, όπου έχουμε εγκαθιδρύσεις κρυπτογραφημένων καναλιών επικοινωνίας, π.χ., μέσω του πρωτοκόλλου TLS (Transport Layer Security). Οι Υποδομές Δημοσίου Κλειδιού λειτουργούν ιεραρχικά, υπό την έννοια ότι υπάρχει μια Αρχή Πιστοποίησης που όλοι εμπιστευόμαστε ή ενδιάμεσες Αρχές Πιστοποίησης τις οποίες εμπιστεύονται άλλες, ανώτερες Αρχές Πιστοποίησης κ.ο.κ. Και κάπου εδώ αποκαλύπτεται ένα σοβαρό πρόβλημα με τις CA: Το μεγάλο τους πλήθος σημαίνει ότι δεν μπορούμε να εμπιστευόμαστε καθεμία εξ αυτών, ενώ και κατά καιρούς έχουν εκδοθεί ψευδεπίγραφα πιστοποιητικά για web sites (βλ., π.χ., http://bit.ly/fakesslcerts).

Τώρα, ειδικά στον κόσμο των PGP, GnuPG και άλλων συστημάτων κρυπτογραφίας δημοσίου κλειδιού που συμμορφώνονται με τις προδιαγραφές του OpenPGP, η σχέση δημοσίου πιστοποιητικού και ιδιοκτήτη επικυρώνεται με βάση το μοντέλο του **Web of Trust**. Σε αντίθεση με το ιεραρχικό μοντέλο που ακολουθεί κάθε Υποδομή Δημοσίου Κλειδιού, στο Web of Trust δεν υπάρχουν Αρχές Πιστοποίησης. Η ιδιοκτησία των δημοσίων κλειδιών επαληθεύεται από τους ίδιους τους χρήστες του συστήματος, χωρίς να υπάρχει κάποια Αρχή την οποία εμπιστευόμαστε εκ προοιμίου. Αναλυτικότερα, οι χρήστες έχουν τη δυνατότητα –χωρίς όμως να είναι υποχρεω-

μένοι– να υπογράφουν όσα πιστοποιητικά θεωρούν έγκυρα. Ένα πιστοποιητικό με πολλές υπογραφές μπορούμε να το εμπιστευόμαστε περισσότερο, σε σχέση με ένα άλλο που δεν έχει υπογράψει κανείς.

Το πώς δουλεύει το μοντέλο του Web Of Trust φαίνεται μέσα από ένα απλό παράδειγμα. Ας υποθέσουμε ότι ένας φίλος μου ή έστω γνωστός μου, έχει πάρει το δημόσιο πιστοποιητικό μου. Αν δεν του το έχω δώσει ο ίδιος, επικοινωνώντας μαζί μου βεβαιώνεται ότι εγώ το παρήγαγα κι αν θέλει το υπογράφει κιόλας. Κατ' αυτόν τον τρόπο δηλώνει δημόσια ότι εγγυάται πως το εν λόγω πιστοποιητικό είναι πράγματι δικό μου. Στη συνέχεια, ένας φίλος του φίλου μου, ο οποίος δεν είναι υποχρεωτικά και δικός μου φίλος, επειδή εμπιστεύεται τον κοινό μας φίλο είναι πιθανό να υπογράψει κι εκείνος το πιστοποιητικό μου. Αμέσως αμέσως, υπάρχουν δύο χρήστες που εγγυώνται για την εγκυρότητα του πιστοποιητικού. Λίγο να το επιδιώξω και οι δύο υπογράφοντες θα γίνουν τρεις ή περισσότεροι.

Μαζικές υπογραφές δημοσίων πιστοποιητικών λαμβάνουν χώρα στα λεγόμενα **key-signing parties**, όπου οι παρευρισκόμενοι έχουν μαζί τους ψηφιακά πιστοποιητικά αλλά και τα απαραίτητα αποδεικτικά της ταυτότητάς τους (π.χ., αστυνομικές ταυτότητες), οπότε άλλοι χρήστες μπορούν, αν το επιθυμούν, να υπογράφουν ψηφιακά πιστοποιητικά χωρίς την παραμικρή αμφιβολία για τους κατόχους τους.

Σε σύγκριση με μια Υποδομή Δημοσίου Κλειδιού, το Web of Trust είναι πολύ πιο ανθεκτικό ως μοντέλο. Η επικύρωση της ιδιοκτησίας των δημόσιων πιστοποιητικών είναι μια ευθύνη που δεν εναποτίθεται σε προνομιακούς παίκτες με συγκεντρωμένη δύναμη, αλλά σε όλους όσοι επιθυμούν να την αναλαμβάνουν. Και το Web of Trust όμως έχει τα προβλήματά του. Η συμπεριφορά –ακριβέστερα η απροθυμία ή/και η αμέλεια– των χρηστών, είναι το πιο σημαντικό. Αν κάποιος χάσει το ιδιωτικό του κλειδί και το αντίστοιχο δημόσιο κλειδί αργεί να λήξει ή δεν έχει καν ημερομηνία λήξης, είναι τότε πιθανό να λαμβάνει κρυπτογραφημένα μηνύματα τα οποία δεν θα μπορεί να αποκρυπτογραφεί. Επιπλέον, αν ένας χρήστης δεν έχει φροντίσει για τη δημιουργία πιστοποιητικού ανάκλησης (**revocation certificate**) και χάσει το ιδιωτικό του κλειδί, τότε ποτέ δεν θα είναι σε θέση ν' ανακαλέσει (ακυρώσει) το δημόσιο κλειδί που κυκλοφορεί Εκεί Έξω (ΤΜ). Ειδικά αν το δημόσιο κλειδί δεν λήγει και ο αρχικός του δημιουργός κάποια στιγμή κυκλοφορήσει νέο πιστοποιητικό, καταλαβαίνετε ότι είναι εύκολο να δημιουργούνται παρεξηγήσεις και το όλο σύστημα της κρυπτογραφίας δημοσίου κλειδιού να δυσφημίζεται άδικα. Τελικά, για να δουλεύει το μοντέλο του Web of Trust οι ίδιοι οι χρήστες του οφείλουν να γνωρίζουν τι κάνουν, βεβαίως και να είναι πρόθυμοι να ελέγχουν λεπτομερώς την ιδιοκτησία των δημοσίων πιστοποιητικών. Αυτό εύκολα λέγεται ή γράφεται, πολύ πιο δύσκολα όμως εφαρμόζεται στην πράξη.

Enter Keybase, όπως θα γράφαμε αν το περιοδικό ήταν στην Αγγλική.

Η αξιοπρόσεκτη προσέγγιση του Keybase

Το Keybase είναι ένα σχετικά νέο εγχείρημα που φιλοδοξεί αφενός να αντιμετωπίσει αποτελεσματικά το πρόβλημα της εμπιστοσύνης στις σχέσεις ιδιοκτησίας φυσικών προσώπων και δημόσιων πιστοποιητικών, αφετέρου να διευκολύνει σημαντικά την καθημερινή χρήση της κρυπτογραφίας δημοσίου κλειδιού.

Ο Chris Coyne, ένας εκ των ιδρυτών του Keybase, όταν αποφάσισε να κατεβάσει τον πηγαίο κώδικα του Bitcoin για να τον εξετάσει, θέλησε να ελέγξει και την εγκυρότητα της διανομής. Αυτό θα γινόταν λαμβάνοντας το δημόσιο κλειδί του Gavin Andresen, του ανθρώπου που εμπιστεύεται ο Satoshi Nakamoto, ο ψευδώνυμος δημιουργός του Bitcoin. Προσπαθώντας όμως ο Coyne να κατεβάσει το δημόσιο κλειδί του Andresen από τους keyservers, βρήκε εκατοντάδες κλειδιά που έδειχναν να τον αφορούν. Ποιο θα έπρεπε να επιλέξει;

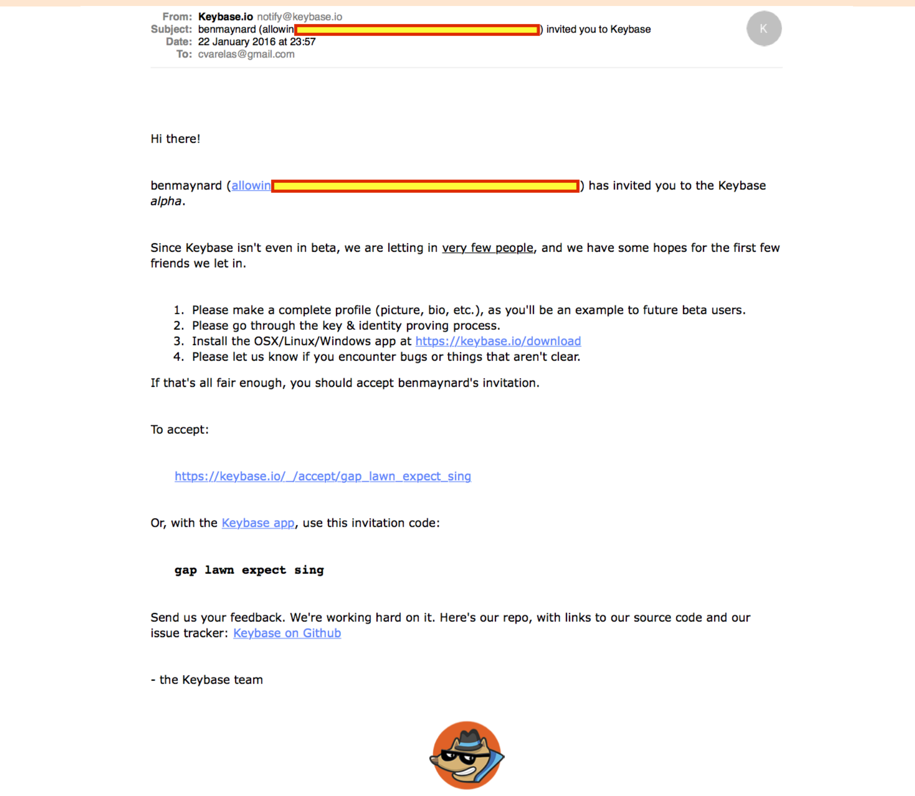
Το Keybase επιτίθεται στο πρόβλημα της εμπιστοσύνης όχι με τη δημιουργία ενός κλειστού, ιδιόκτητου συστήματος ελέγχου, αλλά με βάση τις online παρουσίες των συμμετεχόντων. Όταν μιλάμε για "online παρουσίες" εννοούμε λογαριασμούς σε κοινωνικά δίκτυα, web sites, διευθύνσεις Bitcoin κ.ο.κ. Κάθε χρήστης του Keybase αποδεικνύει τη σχέση του, π.χ., με έναν λογαριασμό στο Twitter ή με έναν δικτυακό τόπο, καταφεύγοντας σε τεχνικές κρυπτογραφίας δημοσίου κλειδιού. Προκειμένου οι υπό συζήτηση αποδείξεις να είναι δυνατές, ο χρήστης είτε πρέπει να διαθέτει ήδη ένα ζεύγος ιδιωτικού/δημοσίου κλειδιού, είτε να δημιουργήσει ένα τέτοιο ζεύγος μέσα από το περιβάλλον του Keybase. Το δημόσιο κλειδί του χρήστη ανεβαίνει στο http://keybase.io –αυτή είναι η διεύθυνση του πρότζεκτ– κι από εκεί και πέρα χρησιμοποιείται, μεταξύ άλλων, ώστε να αποδεικνύεται η σχέση του χρήστη με άλλους online χώρους ή με άλλες ψηφιακές οντότητες (π.χ., διευθύνσεις Bitcoin).

Δείτε τώρα τι γίνεται: Άλλοι χρήστες του Keybase είναι πολύ εύκολο να λαμβάνουν το δημόσιο κλειδί μας, ενώ το σημαντικότερο είναι ότι βεβαιώνονται για την ταυτότητά μας από τις αποδείξεις που έχουμε φέρει εις πέρας. Σε ένα πιο συγκεκριμένο παράδειγμα, κοιτάζοντας κάποιος το προφίλ του γράφοντα στη διεύθυνση https://keybase.io/christos_varelas, βλέπει μεταξύ άλλων μια απόδειξη για το https://twitter.com/subZraw αλλά και αποδείξεις για τα https://colder.xyz, https://parabing.com και http://deltahacker.gr. Δικαιούται, λοιπόν, να είναι σίγουρος για την ταυτότητα του κατόχου του αντίστοιχου δημοσίου κλειδιού, οπότε μπορεί να το πάρει και ν' αρχίσει να μου στέλνει κρυπτογραφημένα email ή αρχεία, βεβαίως και να επαληθεύει το γνήσιο των όποιων υπογραφών μου.

Στο σημείο αυτό θα ισχυριζόταν κάποιος ότι τίποτε δεν εμποδίζει κακόβουλους χρήστες από το να δημιουργούν ψευδεπίγραφα προφίλ στο Keybase. Πράγματι, αν ένας χάκερ έχει αποκτήσει πρόσβαση στο λογαριασμό μου στο Twitter, σίγουρα θα είναι σε θέση να φτιάξει ένα προφίλ με *το δικό μου* ονοματεπώνυμο, *το δικό του* δημόσιο κλειδί και μια απόδειξη για το Twitter (μου). Αν όμως σε ένα προφίλ του Keybase υπάρχουν περισσότερες από μία αποδείξεις, καταλαβαίνουμε ότι η πιθανότητα για έναν χάκερ να έχει αποκτήσει πρόσβαση *σε όλους* τους αντίστοιχους χώρους είναι μηδαμινή. Για παράδειγμα, προκειμένου το προφίλ στο https://keybase.io/christos_varelas να μην έχει δημιουργηθεί από τον γράφο-

ντα, ένας κακόβουλος χάκερ θα έπρεπε να 'χει πρόσβαση σε έναν λογαριασμό στο Twitter, σε άλλον έναν στο Coinbase, καθώς και στους domain name servers τριών διαφορετικών δικτυακών τόπων (colder.xyz, parabing.com και deltahacker.gr).

Προς το παρόν το Keybase λειτουργεί δοκιμαστικά και για να γραφτεί κάποιος στην υπηρεσία χρειάζεται πρόσκληση (invitation) από άλλον, ήδη εγγεγραμμένο χρήστη. Αν ενδιαφέρεστε, μπορούμε να σας εγγυηθούμε ότι δεν θα δυσκολευτείτε να βρείτε μία πρόσκληση. Εμείς, π.χ., στείλαμε ένα απόγευμα email σε χρήστη της υπηρεσίας και την άλλη μέρα είχαμε την πρόσκλησή μας. Αφού γραφτήκαμε στο site και δημιουργήσαμε το προφίλ μας, σε λίγες ώρες μάς είχαν σταλεί άλλες οκτώ προσκλήσεις. Αυτές, εννοείται ότι μπορούμε να τις στείλουμε σε άλλους ενδιαφερόμενους. Δείτε στα screenshots που ακολουθούν μερικά στιγμιότυπα από την ενασχόλησή μας με την υπηρεσία, διαβάστε και τις αντίστοιχες περιγραφές.



From: Keybase.io notify@keybase.io
Subject: benmaynard (allowin[redacted]) invited you to Keybase
Date: 22 January 2016 at 23:57
To: cvarelas@gmail.com

Hi there!

benmaynard (allowin[redacted]) has invited you to the Keybase *alpha*.

Since Keybase isn't even in beta, we are letting in very few people, and we have some hopes for the first few friends we let in.

1. Please make a complete profile (picture, bio, etc.), as you'll be an example to future beta users.
2. Please go through the key & identity proving process.
3. Install the OSX/Linux/Windows app at https://keybase.io/download
4. Please let us know if you encounter bugs or things that aren't clear.

If that's all fair enough, you should accept benmaynard's invitation.

To accept:

https://keybase.io/_/accept/gap_lawn_expect_sing

Or, with the Keybase app, use this invitation code:

gap lawn expect sing

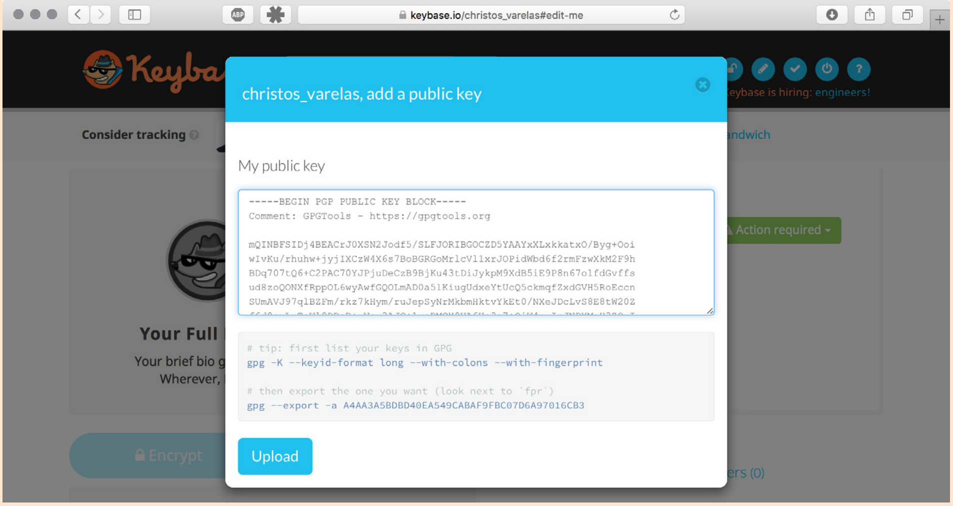
Send us your feedback. We're working hard on it. Here's our repo, with links to our source code and our issue tracker: Keybase on Github

- the Keybase team

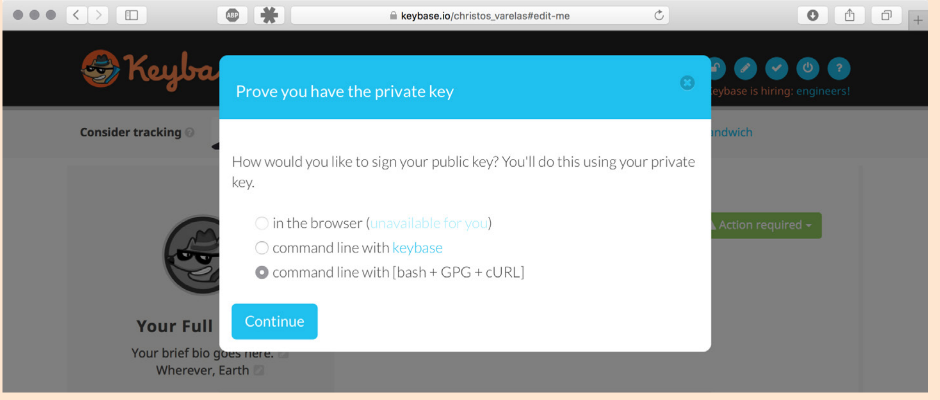
Προς το παρόν το Keybase δεν είναι καν σε φάση beta και για να γραφτείτε στο site θα χρειαστείτε πρόσκληση από ήδη εγγεγραμμένο χρήστη. Δεν δυσκολευτήκαμε να βρούμε πρόσκληση και δεν θα δυσκολευτείτε ούτε κι εσείς. Απλά, ξεκινήστε από το https://keybase.io κι επισκεφτείτε το προφίλ κάποιου χρήστη. Αν διαθέτει προσκλήσεις, το γεγονός αυτό θα αναφέρεται ευκρινώς σε ένα πράσινο πλαίσιο. Κατεβάστε το δημόσιο κλειδί του χρήστη και στείλτε του ένα κρυπτογραφημένο email, στο οποίο ευγενικά θα του ζητάτε να σας στείλει μια πρόσκληση. Πιθανότατα θα έχετε κι άλλες δυνατότητες για επικοινωνία, π.χ., μέσω Twitter, πιστεύουμε όμως ότι με την αποστολή κρυπτογραφημένου email θα τραβήξετε την προσοχή του και θα σου μπεράνει ότι λαμβάνετε σοβαρά υπόψη την κρυπτογραφία δημοσίου κλειδιού. Έτσι, θα είναι πιο δύσκολο για εκείνον να σας αρνηθεί μια πρόσκληση.



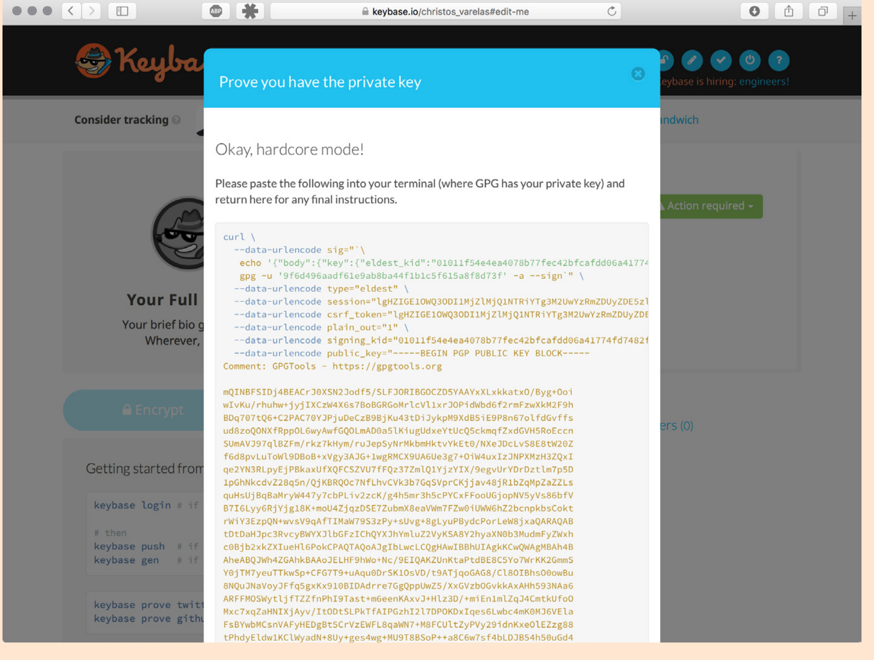
Μετά το login στην υπηρεσία το προφίλ μας είναι άδειο και φυσικά δεν υπάρχει καμία απόδειξη για τις online παρουσίες μας. Η πρώτη μας ενέργεια θα πρέπει να είναι η προσθήκη ενός εκ των δημοσίων κλειδιών που διαθέτουμε. Αν δεν έχουμε κάποιο ζεύγος ιδιωτικού/δημοσίου κλειδιού, μπορούμε να δημιουργήσουμε ένα μέσα από το web interface του Keybase.



Διαθέταμε ήδη ζεύγος ιδιωτικού/δημοσίου κλειδιού, δεν είχαμε λόγο να δημιουργήσουμε νέο, συνεπώς επιλέξαμε να ανεβάσουμε το δημόσιο κλειδί μας στο Keybase. Παρατηρήστε ότι μας παρέχονται και οδηγίες για το πώς μπορούμε να εξάγουμε το δημόσιο κλειδί μας από τη γραμμή εντολών σε μορφή ASCII. Αφού το εξάγουμε –όχι κατ' ανάγκη με τον υποδεικνυόμενο τρόπο–, επιλέγουμε όλο το κείμενο, το μεταφέρουμε στο πλαίσιο My public key και πατάμε στο κουμπί Upload.



Μετά το ανέβασμα του δημοσίου κλειδιού οφείλουμε να το υπογράψουμε με χρήση του ιδιωτικού μας κλειδιού. Επιλέγουμε να το πράξουμε ακολουθώντας την 3η από τις τρεις διαθέσιμες μεθόδους, δηλαδή από τη γραμμή εντολών του BASH shell και με τη βοήθεια των εργαλείων gpg και curl. Σε συστήματα Linux/BSD/OS X, όλα όσα χρειαζόμαστε μάλλον θα είναι ήδη εγκατεστημένα. Αν κάτι απουσιάζει, εύκολα θα μπορέσουμε να το προσθέσουμε. Σε περίπτωση που δουλεύουμε σε περιβάλλον Windows, το gpg διατίθεται μέσω του πακέτου Gpg4win (https://www.gpg4win.org). Υπάρχει εξάλλου και binary του curl, στο http://www.confusedbycode.com/curl/#downloads. Μια εναλλακτική είναι να εγκαταστήσουμε τον επίσημο client του Keybase για Windows (https://keybase.io/download) και για την υπογραφή του δημοσίου κλειδιού να χρησιμοποιήσουμε τη 2η από τις τρεις διαθέσιμες μεθόδους.



Στο τερματικό μας θα αντιγράψουμε όλο το περιεχόμενο του πλαισίου που μας έχει παρουσιαστεί. Αρκεί ένα κλικ μέσα στο πλαίσιο κι όλο το περιεχόμενο θα επιλεχθεί, οπότε είναι εύκολο να το αντιγράψουμε στο clipboard (copy) και μετά να το μεταφέρουμε σε ένα παράθυρο τερματικού (paste). Όλο αυτό το κείμενο δείχνει τρομακτικό. Ανόμως προσέξουμε λίγο, βλέπουμε ότι με τη βοήθεια του curl και μέσω του API που παρέχει το Keybase, αυτό που κάνουμε είναι ότι υπογράφουμε τοπικά το δημόσιο κλειδί μας και την υπογραφή την ανεβάζουμε στο Keybase.

```
> KlnzRLwC0N6kXwXIu8fcEZcVOFmlUdnRN+MNpW1wB1iziU7JDuxGzcIw3cpB40Gf
> zO5CFXHybRQ9Dd5QK7FtIS8jos6NuDQkp0TBwNtcMPq8m+xSplZqalKOKausMN7w
> LcOvSne7xwQW7MlgOVluUxjTDv4NW8M5wwyurB7abgIAP/XGwr54murDLx2y/J86
> rifsiM3BJJr8w8G8NKrxLhVh0m0WuqCyJsCNgVvC2nVtg2MrnsbG3Ap0uEj0JuUt
> Etj75Z6YUSavSKCrgR2HkJb+OezGZbmAXt4nfH7bpXbdzKpSBXiN+niZLN9fd+e6
> 1hp6ul6svfEy0nw5uSwtfIgyAiFm0X5Oy7OVYqUBNQL3W0KIklGQd1V6Zonv2yMM
> QOWKSUTaRaMvLcRajKHwUQWcnC6KvsoX3Y+p7G/vNxvd3wvgEQwfAVceGkF7o7Z+
> GQjNzwWPfrabd9uY7w4s5p20UEOBxuXAw9g8FsVE3v7CRiSMviY+m/+TmrNa7fII
> ECwClH4bCZ3HcBhhasBv+pDOjM/pWzvGKgcAuYqtBWsN5dyrhAJGTAa3eIj71l57
> YJ5oVTD+9+DISdld+QQGTUQOmd0oJjE6w2mdTKbE5cVG9EIZNcz93RTIYFqtSfP3
> iokCHAQTAQoABgUCVLDlMQAKCRBbcyQ99Nx7hDL0D/4kZDxdYw/cqScDnsxHwftZ
> /AUnJW0tLIBi4r1IwN90U2hksOweZj+i82Fwg4Pd/R3YipKAgIe8ofkOJjVMntPh
> 6XXG5QLiQn/L+lIKevh2Y+c4WEXzF11lipfaElkipDR55XGkdaALyRS1gZkg75rc
> gVjKRtLjaIeys58zpxqlm5oSKFEwl5NWdPj4V7whoqVNcGcDav3rPjdSI628+/l6
> Ghkn2pZTcJvedLxfZE9uMorU5qdr0MBL1Eprcx84W60xR95MtI8xq3dAoV15EL9Z
> vTIBh3WkcnJuCGngWQp19KAmRgfVcVnj62y/Pc/qCzi0nkMud4ekMkMMaiCp0mzD
> pDnH5GocpJyVyXAq6238WvIe3mLISBkY4nmJSweqMJrBtBeid94CKECvDqi5EyBR
> nKKTgWZCNFEuHuqRDF+boIkCPjIEqe6BSq3fd/BzhAm06FV9CP6/qOZqFbiptT9
> Qodhya/cTaTMJUut7ZWldNgUmJ8lNHYUzKtgNKBPRRpGLaI0/qK0ajlml2tgTi+e
> laFqt0ll9U1/AR6qooW72GPpkQhPkSO8xJbcT9ChIx5BdzMdYmbv22nRpwdbu1iM
> f13LNE0lsJKLoXYMkwyCnKWObMKtSOs2NaxFvTd8KdEe7JCzp1o2do0kg6xovc+n
> Awv1i7nm48IHq2Omp7cSpokCHAQTAQoABgUCVLDlYwAKCRAOOcMI77ovmMdfEACF
> 4aedSAvDexVI0JZuN4RKIgGHNPWTtbKk7tLytyhVcmY0Yh0hI4k9D/gfiGFVkKlY
> bdfF+rFVi/GKpOXZWixKv3j2mwXE2+cXo6SoV/nfx9BfyJXfiEj7g3mFnLbPeSFK
> zKYLcgLK3zxgzNOpOJm49dRBoRdCEU0dlEYw1J8OCw93OZ9UvuVEWJXaBaIuhbFW
> KNN/Ktnswo+MPzhgK7DhHsKGMUox97xQhrScLl4mtOUEJzIqTW6Opf6UOk9EQWh1
> D6hctd7760KhaoEVYR0kEK3pbFdFxvo/Qc5z0DMQfMjRZcwwL2K8fFr7Rxogm9Dd
> lBo1+r8Xpgd1jTaSGhTKG+8uZhnsCOyEPVL57VhXhjGYzoEIWQB1RSl/x/KK31S1
> wAkSI3qNki4+qJzUC5gG2yR0s95Mi+/JtYqs22fGAhWTwhj5h2yZTTGf3DvncbmV
> +dp+VzAe6FvhvqY9b6p1/SEBaZKlpMKZbAf67uYOXuaxA+Xszw1RIhYD2PfwP5dq
> HRZa1BOHnHlvbFKfVo7h52ziRxRMQF83/omPekW7ikheJ6TstkV4MAPfPNyexVBW
> KrHoFpRzn/gXa4Xx/nX08jaSFmYg5IIGOQcd/hwug1dajBrY77g82klSl4k/u8sl
> n8K8CgKMLImFDBgnRop166fl/kWU0uqQeL5NPq+R+g==
> =yDQT
> -----END PGP PUBLIC KEY BLOCK-----" \
>   --data-urlencode is_primary="true" \
>   --data-urlencode sig_required="true" \
>   https://keybase.io/_/api/1.0/key/add.json

You need a passphrase to unlock the secret key for
user: "Christos Varelas (Parabinger) <christos.varelas@colder.xyz>"
4096-bit RSA key, ID A8F8D73F, created 2014-12-10

Enter passphrase:
```

Έχουμε μεταφέρει το κείμενο στο τερματικό κι έχουμε εκτελέσει τη σχετική εντολή. Πριν δημιουργηθεί η υπογραφή καλούμαστε να πληκτρολογήσουμε το passphrase που προστατεύει το ιδιωτικό μας κλειδί.

```
> -----END PGP PUBLIC KEY BLOCK-----" \
>    --data-urlencode is_primary="true" \
>    --data-urlencode sig_required="true" \
>    https://keybase.io/_/api/1.0/key/add.json

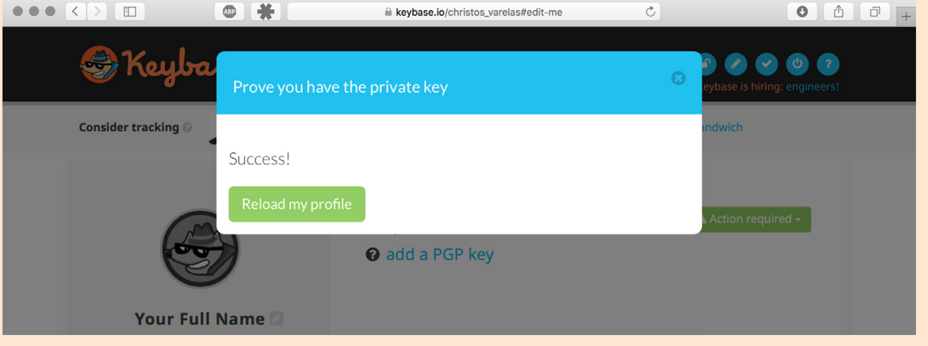
You need a passphrase to unlock the secret key for
user: "Christos Varelas (Parabinger) <christos.varelas@colder.xyz>"
4096-bit RSA key, ID A8F8D73F, created 2014-12-10

Success!
========

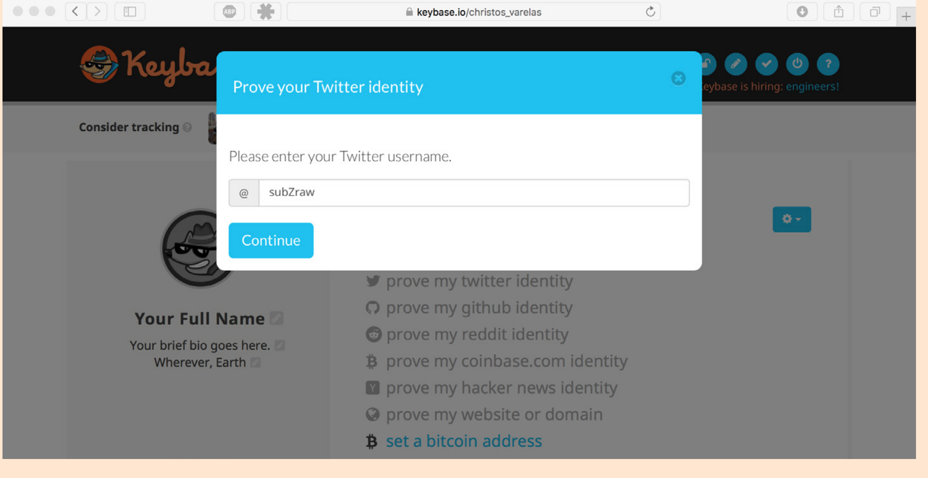
Please refer to your browser session for more information.

(Signature id: b0518dda47d151ec4a1c99df09e63047c0f8adedbe52138762f962f4fe40971a0f)
mbpr15:~ cvar$
```

Ωραία! Δημιουργήθηκε η υπογραφή, ανέβηκε και στο Keybase. Από το τερματικό, παροτρυνόμαστε να κοιτάξουμε στον web browser.



Το μήνυμα στον browser δεν αφήνει καμία αμφιβολία: Μόλις αποδείξαμε ότι έχουμε στην κατοχή μας το ιδιωτικό κλειδί που αντιστοιχεί στο δημόσιο, το οποίο πριν λίγο ανεβάσαμε στο Keybase.



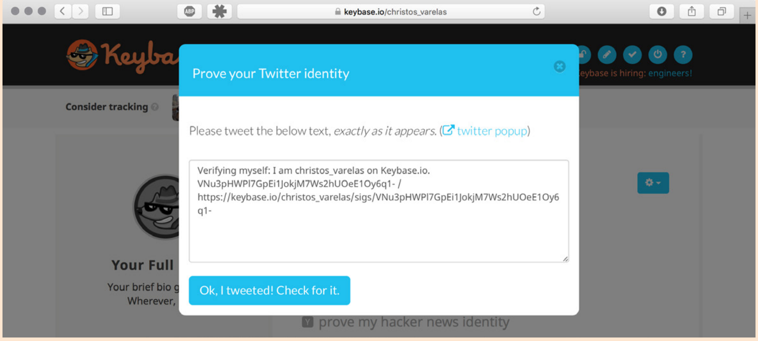
Αναφορικά με τις παρουσίες μας online, η πρώτη που θελήσαμε να αποδείξουμε ότι είναι πράγματι δική μας αφορούσε στο Twitter. Η διαδικασία ξεκινά πληκτρολογώντας το Twitter handle μας.

```
mbpr15:~ cvar$ curl \
>   --data-urlencode sig="`\
>    echo '{"body":{"key":{"eldest_kid":"01011f54e4ea4078b77fec42bfcafdd06a41774fd7482f68c6f47578138f2fe23f9f0a","fingerprint":"9f6d496aadf61e9ab8ba44f1b1c5f615a8f8d73f","host":"
keybase.io","key_id":"b1c5f615a8f8d73f","kid":"01011f54e4ea4078b77fec42bfcafdd06a41774fd7482f68c6f47578138f2fe23f9f0a","uid":"a59d782526e24554ba873e0c4fd52d19","username":"christ
os_varelas"},"service":{"name":"twitter","username":"subZraw"},"type":"web_service_binding","version":1},"ctime":1453502530,"expire_in":157680000,"prev":"aaa9f4722f85fb2ac891f021
4abe7e55755087c66c5474633fec4040321e9383","seqno":2,"tag":"signature"}' | \
>    gpg -u '9f6d496aadf61e9ab8ba44f1b1c5f615a8f8d73f' -a --sign`" \
>   --data-urlencode type="web_service_binding.twitter" \
>   --data-urlencode session="lgHZIGE1OWQ3ODI1MjZlMjQ1NTRiYTg3M2UwYzRmZDUyZDE5zlaiq/HOAeEzgNkgOTA3NTM3NGMyZTIxMzVlM2FkZmM3YzllYTA0Y2ZiMDLEIDt8ztT/WGGVaWNuSsd/tab6KxWYOGz/JgxwbZQy
LE7y" \
>   --data-urlencode csrf_token="lgHZIGE1OWQ3ODI1MjZlMjQ1NTRiYTg3M2UwYzRmZDUyZDE5zlaiq/HOAAFRgMDEIIrrfDHKczmLq0D4750STjN2y7msQtOQJnneueKKVRAW" \
>   --data-urlencode plain_out="1" \
>   --data-urlencode signing_kid="01011f54e4ea4078b77fec42bfcafdd06a41774fd7482f68c6f47578138f2fe23f9f0a" \
>   --data-urlencode remote_username="subZraw" \
>   https://keybase.io/_/api/1.0/sig/post.json

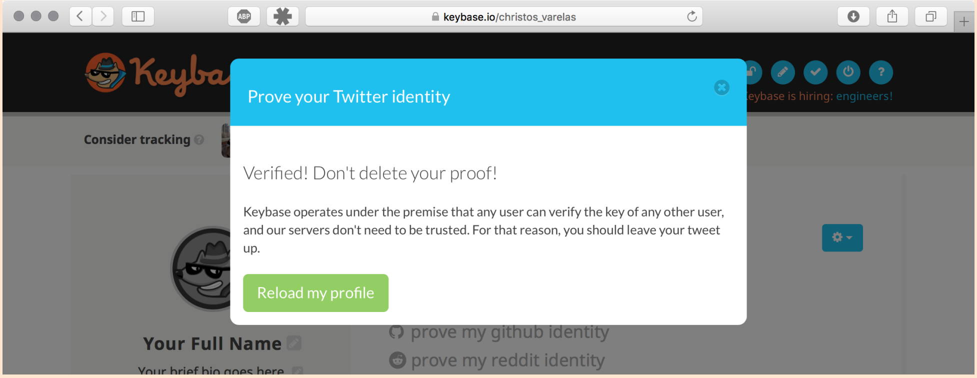
You need a passphrase to unlock the secret key for
user: "Christos Varelas (Parabinger) <christos.varelas@colder.xyz>"
4096-bit RSA key, ID A8F8D73F, created 2014-12-10

Enter passphrase:
```

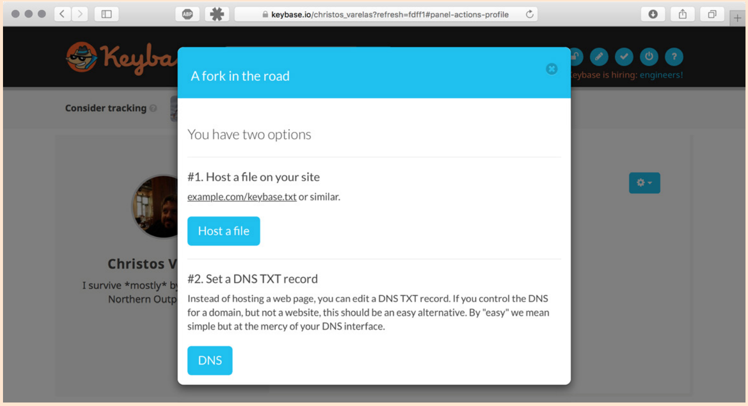
Για άλλη μια φορά επιλέξαμε να κάνουμε την απόδειξη από το τερματικό και με τα εργαλεία curl και gpg. Το Keybase παρέθεσε τον σχετικό κώδικα, τον μεταφέραμε σε ένα τερματικό και πριν υπογράψουμε ένα επιθυμητό κείμενο (tweet) που δημιουργεί το Keybase, απλά πληκτρολογήσαμε το passphrase του ιδιωτικού μας κλειδιού.



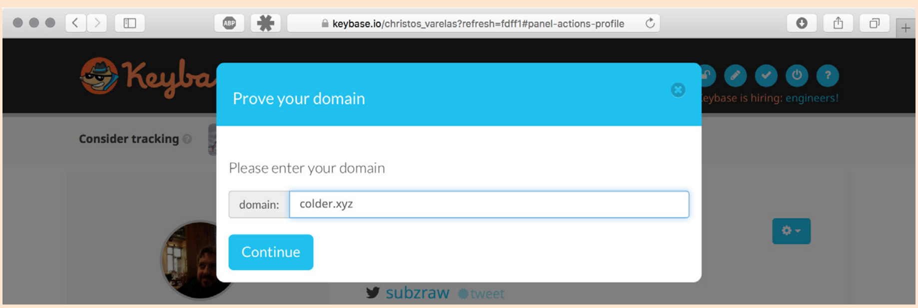
Στη συνέχεια κάνουμε ένα tweet, ακριβώς με το περιεχόμενο που μας δίνεται από το Keybase. Μετά το tweet λέμε στο Keybase να το ελέγξει.



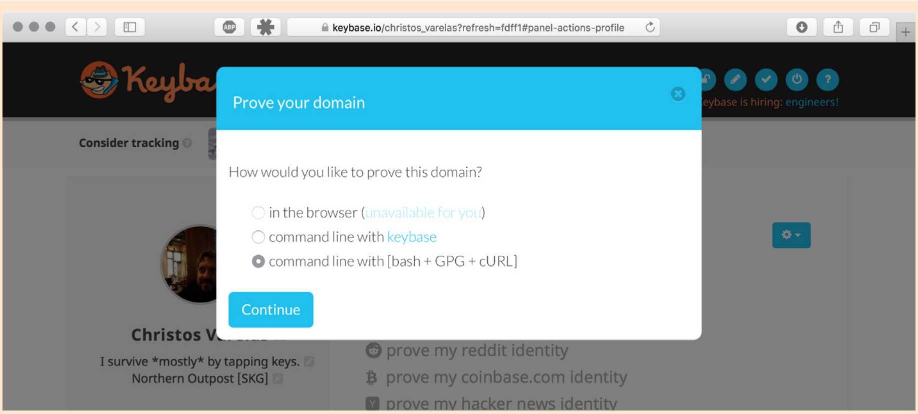
Με χρήση του δημοσίου μας κλειδιού, το keybase αποκρυπτογραφεί την ψηφιακή υπογραφή που ανεβάσαμε. Το plaintext που παίρνει ως αποτέλεσμα το συγκρίνει με το tweet που μόλις κάναμε. Αν τα δύο κείμενα ταυτίζονται, τότε αποδεικνύεται ότι πράγματι ελέγχουμε τον αντίστοιχο λογαριασμό στο Twitter. Προσέξτε: Παροτρυνόμαστε να μη διαγράψουμε το tweet, αφού άλλοι χρήστες θα θέλουν να ελέγχουν από μόνοι τους (με τη βοήθεια ενός εργαλείου σαν το keybase, https://keybase.io/download) ότι είμαστε οι κάτοχοι του αντίστοιχου λογαριασμού. Όταν λέμε "από μόνοι τους", εννοούμε ότι ο έλεγχος γίνεται χωρίς ο ενδιαφερόμενος να εμπιστεύεται τι λέει ο Keybase server.



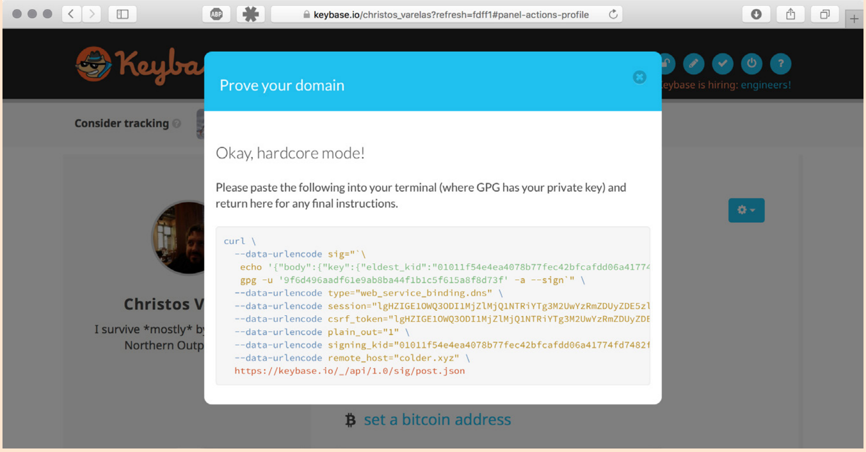
Έχουμε τρία διαφορετικά sites υπό τον έλεγχό μας, οπότε σκεφτήκαμε ότι καλό είναι να παραθέσουμε ισάριθμες αποδείξεις, μία για κάθε site. Η επαλήθευση της ιδιοκτησίας ενός site γίνεται με δύο τρόπους: είτε με το ανέβασμα ενός αρχείου στον κατάλληλο κατάλογο του web server, είτε με την προσθήκη ενός TXT record στον name server που είναι υπεύθυνος για το domain του site. Μια κι έχουμε πρόσβαση στους name servers των sites, αποφασίσαμε να ακολουθήσουμε τη δεύτερη μέθοδο.



Στο αμέσως επόμενο βήμα παρέχουμε το domain name του site. Ξεκινήσαμε με την επαλήθευση για το colder.xyz.



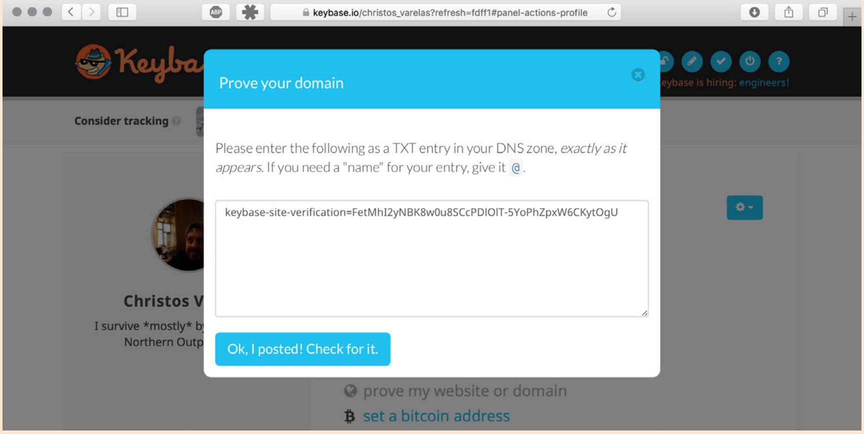
Το περιεχόμενο του TXT record που πρόκειται να μας δοθεί θα το υπογράψουμε και την ψηφιακή υπογραφή θα την ανεβάσουμε στο Keybase. Για άλλη μια φορά επιλέγουμε τη μέθοδο των BASH, gpg και curl.



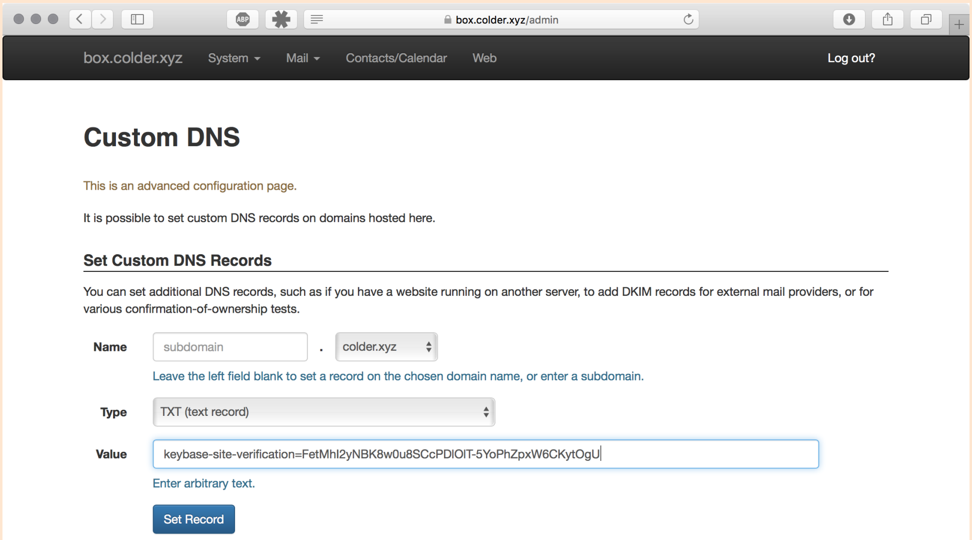
Η συνέχεια είναι η αναμενόμενη: αντιγράφουμε τον κώδικα που μας παρέχει το Keybase...



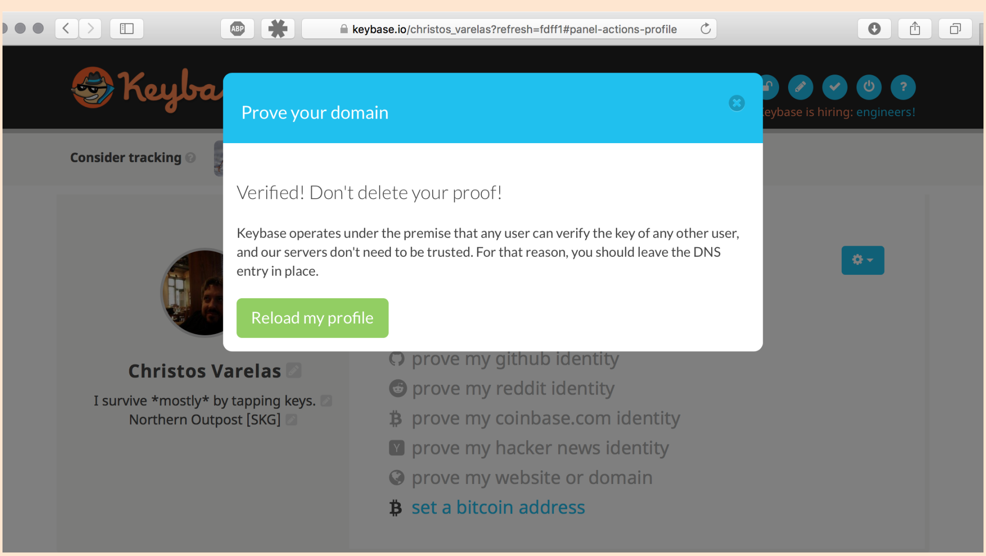
...τον μεταφέρουμε σε ένα τερματικό και τον εκτελούμε. Όπως βλέπετε, η ψηφιακή υπογραφή μπήκε κι ανέβηκε επιτυχώς στο Keybase.



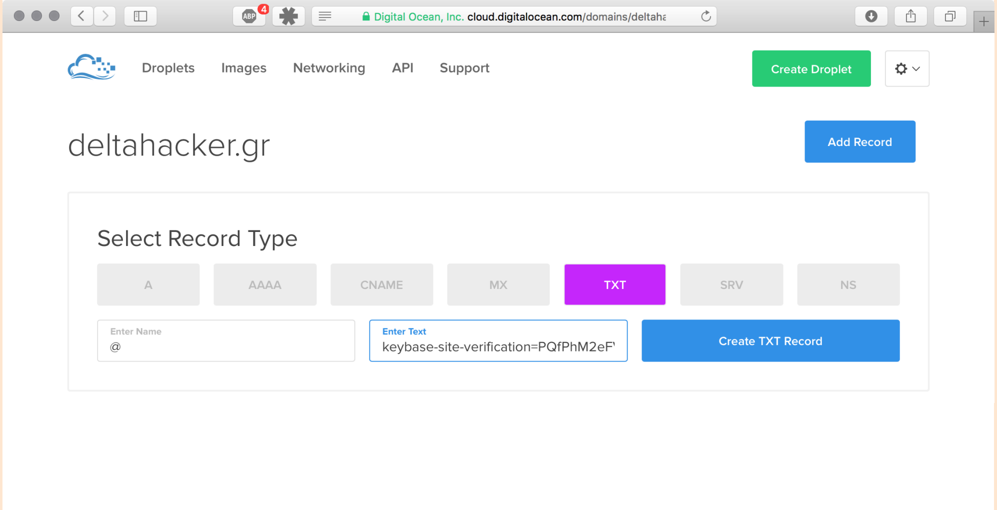
Αντιγράφουμε ως έχει το περιεχόμενο που το Keybase παραθέτει στο σχετικό πλαίσιο. Αυτό είναι το περιεχόμενο του νέου TXT record που θα φτιάξουμε.



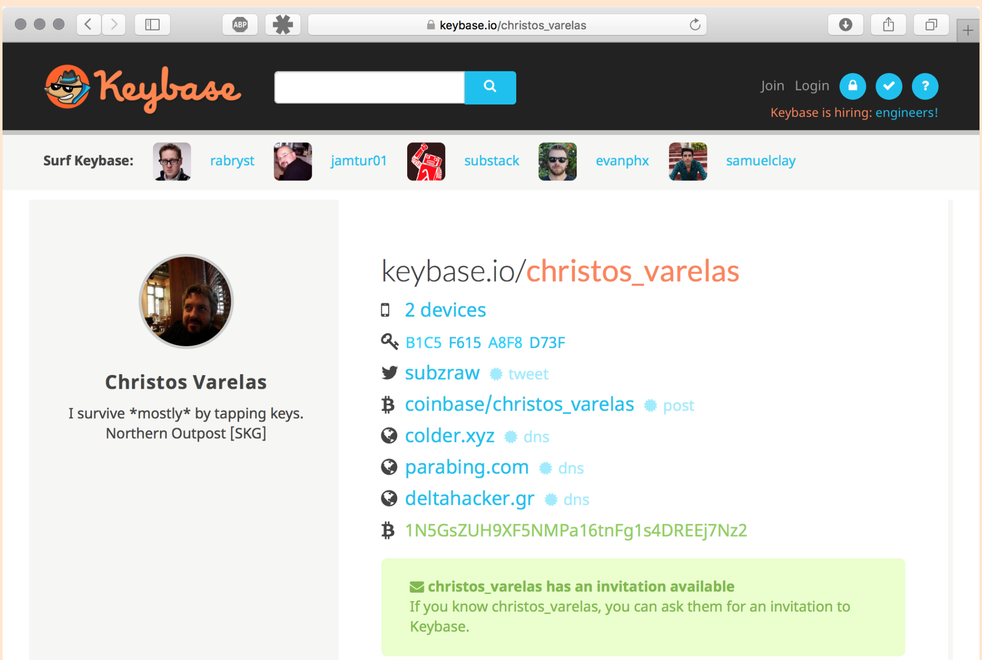
Ο name server του colder.xyz είναι εκείνος του Mail-in-a-Box (http://deltahacker.gr/?p=14096) και η διαχείρισή του πραγματοποιείται μέσα από ένα λιτό κι εύχρηστο web panel. Στο στιγμιότυπο βλέπουμε τη δημιουργία ενός νέου record τύπου TXT [Type = TXT (text record)] για το colder.xyz (Name = colder.xyz). Το περιεχόμενο του νέου record είναι αυτό που μας έδωσε το Keybase. Ευλαβικά, το παραθέτουμε στη θυρίδα Value.



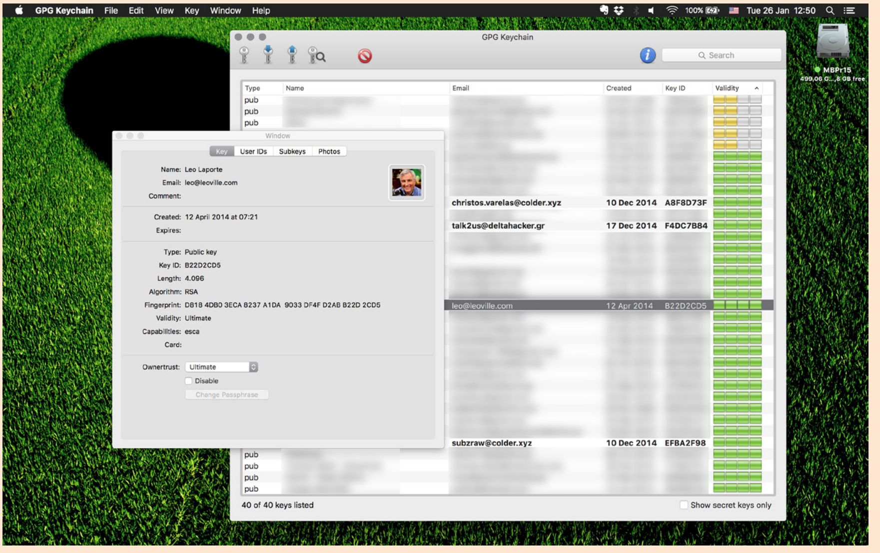
Μετά από λίγο το Keybase θα είναι σε θέση να διαβάσει το νέο TXT record από τον name server του colder.xyz, επομένως θα επαληθεύσει την ιδιοκτησία του domain από μέρους μας. Όπως και με την περίπτωση του Twitter, έτσι και τώρα παροτρυνόμαστε να μη διαγράψουμε το TXT record.



Στα web panel κάποιων name servers, όταν ορίζουμε ένα νέο TXT record τότε ως όνομα του δίνουμε το "@" (χωρίς τα εισαγωγικά). Το Keybase φροντίζει και μας πληροφορεί σχετικά και τη συμβουλή του την ακολουθήσαμε κατά την επαλήθευση της ιδιοκτησίας του deltahacker.gr, όπου ο name server του domain παρέχεται από τη Digital Ocean.



Ιδού το προφίλ του γράφοντα, με όλες τις αποδείξεις που έχει πραγματοποιήσει. Ένας γρήγορος τρόπος προκειμένου να εισάγετε το δημόσιο κλειδί του στο keyring σας, είναι ν' ανοίξετε ένα τερματικό και να πληκτρολογήσετε "curl https://keybase.io/christos_varelas/key.asc | gpg --import" (χωρίς τα εισαγωγικά).



Διαχείριση των δημοσίων κλειδιών του keyring μας, μέσα από το περιβάλλον του GPG Keychain manager (για OS X). Τα περισσότερα κλειδιά ανήκουν σε αναγνώστες του περιοδικού αλλά κάποια τα έχουμε πάρει από το Keybase. Ελέγχοντας τις αποδείξεις ορισμένων χρηστών με τη βοήθεια του εργαλείου ονόματι keybase (π.χ., "keybase id leolaporte"), είναι πιθανό να διαπιστώσετε ότι κάποιες αποτυγχάνουν. Αυτό μάς συνέβη με το προφίλ του Leo Laporte. Αν γνωρίζετε τον υπό έλεγχο χρήστη, δεν είναι καθόλου κακή ιδέα να τον ενημερώσετε αφού οι όποιες αποτυχίες είναι πιθανό να οφείλονται σε δικές του ενέργειες (ίσως, π.χ., να αφαίρεσε ένα tweet ή κάποιο TXT record).

Ενδιαφέρουσες δυνατότητες

Στην παρούσα μορφή του, το Keybase μοιάζει με ένα directory χρηστών και αντίστοιχων δημοσίων κλειδιών, το οποίο μπορούμε να εμπιστευόμαστε. Αν μη τι άλλο, λοιπόν, από το keybase.io έχουμε τη δυνατότητα λήψης δημοσίων κλειδιών άλλων χρηστών, με τους οποίους θα έχουμε την επιλογή για ισχυρά κρυπτογραφημένη επικοινωνία. Από το δικτυακό τόπο του πρότζεκτ και συγκεκριμένα από το https://keybase.io/download διατίθεται και το keybase, ένα εργαλείο για τη γραμμή εντολών που διευκολύνει την πραγματοποίηση των αποδείξεων, μας επιτρέπει ν' αναζητάμε άλλους χρήστες, να τους ακολουθούμε (track), να κρυπτογραφούμε/αποκρυπτογραφούμε μηνύματα κ.ο.κ.

Δεν σας κρύβουμε ότι αναμένουμε με ιδιαίτερο ενδιαφέρον τις αντιδράσεις προγραμματιστών και χρηστών, ειδικά μετά το αναμενόμενο άνοιγμα του Keybase στο ευρύ κοινό. Το ομώνυμο εργαλείο για τη γραμμή εντολών αλλά και το ίδιο το web site, αποτελούν δύο παραδείγματα clients για την υπηρεσία. Οι developers έχουν τη δυνατότητα να την παντρέψουν με υπάρχουσες εφαρμογές, καθιστώντας έτσι την κρυπτογραφία δημοσίου κλειδιού προσβάσιμη σε όλους. Κι αν όχι σε όλους, έστω σε μεγάλο ποσοστό χρηστών που δεν υπάρχει περίπτωση να κάνουν κτήμα τους το μαθηματικό ή/και το τεχνικό υπόβαθρο για την ορθή, καθημερινή χρήση της κρυπτογραφίας δημοσίου κλειδιού. Πιστεύουμε ότι ένα τεράστιο βήμα προς αυτή την κατεύθυνση θα ήταν η ενσωμάτωση των λειτουργιών του Keybase σε δημοφιλείς email ή/και chat clients. Μπορεί να κάνουμε λάθος – αλλά αυτή τη φορά ειλικρινά θα μας πειράξει αν κάνουμε.

Skill: Intermediate
Tags: Certification Authority, TLS, certificates, nginx

Δωρεάν πιστοποιητικά για όλους, εδώ και τώρα!

Οι δικαιολογίες τελείωσαν. Το εγχείρημα Let's Encrypt είν' εδώ, ήδη σε φάση beta, και όλοι οι ενδιαφερόμενοι μπορούν να έχουν δωρεάν πιστοποιητικό TLS για το site τους. Μέχρι κι εμείς βάλαμε, οπότε έχετε μία δικαιολογία λιγότερη για να το καθυστερήσετε κι άλλο.

του Χρήστου Βαρελά

Η υπηρεσία του Let's Encrypt (https://letsencrypt.org) παρέχεται από το μη-κερδοσκοπικό οργανισμό Internet Security Research Group και υποστηρίζεται από μεγάλα ονόματα όπως Electronic Frontier Foundation, Mozilla Foundation και Cisco Systems. Η αποστολή του όλου πρότζεκτ είναι η εύκολη και δωρεάν παροχή πιστοποιητικών X.509 για την υλοποίηση κρυπτογραφημένων συνδέσεων TLS. Με πολύ απλά λόγια, χάρη στο Let's Encrypt μπορούμε να έχουμε HTTPS για τα sites μας με μηδενικό κόστος, για πάντα. Το HTTPS δεν είναι κάτι για το οποίο μπορούμε ν' αδιαφορούμε – και δεν έχει σημασία αν στο site μας αποστέλλονται ευαίσθητα δεδομένα χρηστών ή όχι. Μεγάλες εταιρείες κι οργανισμοί όπως η Google και η Mozilla φιλοδοξούν να καταστήσουν τις συνδέσεις HTTP απομεινάρι του παρελθόντος. Οι συνδέσεις HTTPS θα είναι οι προκαθορισμένες στο web και το εγχείρημα του Let's Encrypt ανοίγει το δρόμο προς αυτή την κατεύθυνση.

Στις αρχές του περασμένου Δεκεμβρίου το Let's Encrypt ως Certification Authority (CA) μπήκε σε φάση beta. Προς το παρόν, αυτόματη λήψη πιστοποιητικού και ρύθμιση παρέχεται μόνο για τους web servers με Apache. Όσοι τρέχουν άλλον web server εξακολουθούν να έχουν το δωρεάν πιστοποιητικό τους, απλά χρειάζεται να κάνουν και μερικές επεμβάσεις στο κατάλληλο αρχείο ρυθμίσεων του web server. Εμείς, π.χ., είμαστε υπεύθυνοι για δύο sites με nginx: το deltahacker.gr και το parabing.com. Και στα δύο έχουμε ήδη εγκαταστήσει πιστοποιητικό TLS και στο παρόν άρθρο δείχνουμε τη διαδικασία που ακολουθήσαμε, αναλυτικά και βήμα προς βήμα.

Προετοιμασία

Η παρουσίασή μας γίνεται από το VPS του parabing.com, το οποίο τρέχει Ubuntu Server 14.03 LTS 64bit (παρεμπιπτόντως, το ίδιο OS έχει και το VPS του deltahacker.gr). Για τη λήψη και την εγκατάσταση πιστοποιητικών TLS, το πρότζεκτ του Let's Encrypt παρέχει το εργαλείο letsencrypt. Προς το παρόν δεν διατίθεται έτοιμο κάποιο πακέτο DEB για Ubuntu, όμως εύκολα μπορούμε να εγκαταστήσουμε το letsencrypt από το GitHub. Ξεκινάμε αναβαθμίζοντας τα repositories του Ubuntu Server, καθώς και όποια πακέτα χρειάζονται εγκατάσταση:

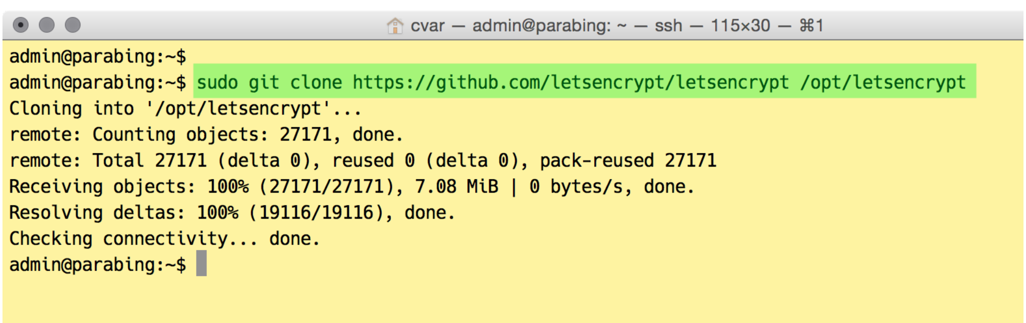
```
admin@parabing:~$ sudo apt-get update
admin@parabing:~$ sudo apt-get upgrade
```

Προχωράμε με την εγκατάσταση του git – αν δεν είναι ήδη παρόν στο σύστημά μας:

```
admin@parabing:~$ sudo apt-get install git
```

Αμέσως μετά κλωνοποιούμε τον κατάλογο του Let's Encrypt από το GitHub, στο /opt/letsencrypt:

```
admin@parabing:~$ sudo git clone https://github.com/letsencrypt/letsencrypt /
opt/letsencrypt
```



Επιτυχής κλωνοποίηση του letsencrypt από το GitHub. Προς το παρόν δεν διατίθεται έτοιμο πακέτο για το Ubuntu, αλλά καθόλου δεν πτοούμαστε.

Λήψη πιστοποιητικού

Προκειμένου το Let's Encrypt να επιβεβαιώσει ότι είμαστε οι κάτοχοι του domain για το οποίο ζητάμε πιστοποιητικό (στο παράδειγμά μας το εν λόγω domain είναι το parabing.com), στον name server για το domain οφείλουμε να έχουμε ένα A record που να δείχνει στον server ή στο VPS του site (περισσότερα περί DNS διαβάστε στο https://deltahacker.gr/?p=14749). Αν έχετε ένα λειτουργικό site με το domain του,

τότε μάλλον δεν θα κάνετε κάποια αλλαγή στο DNS. Επειδή εξάλλου το Let's Encrypt χρειάζεται να σηκώσει, προσωρινά, έναν δικό του HTTP server ο οποίος θα ακούει στο port 80, προς το παρόν πρέπει να σταματήσουμε τον nginx:

```
admin@parabing:~$ sudo service nginx stop
```

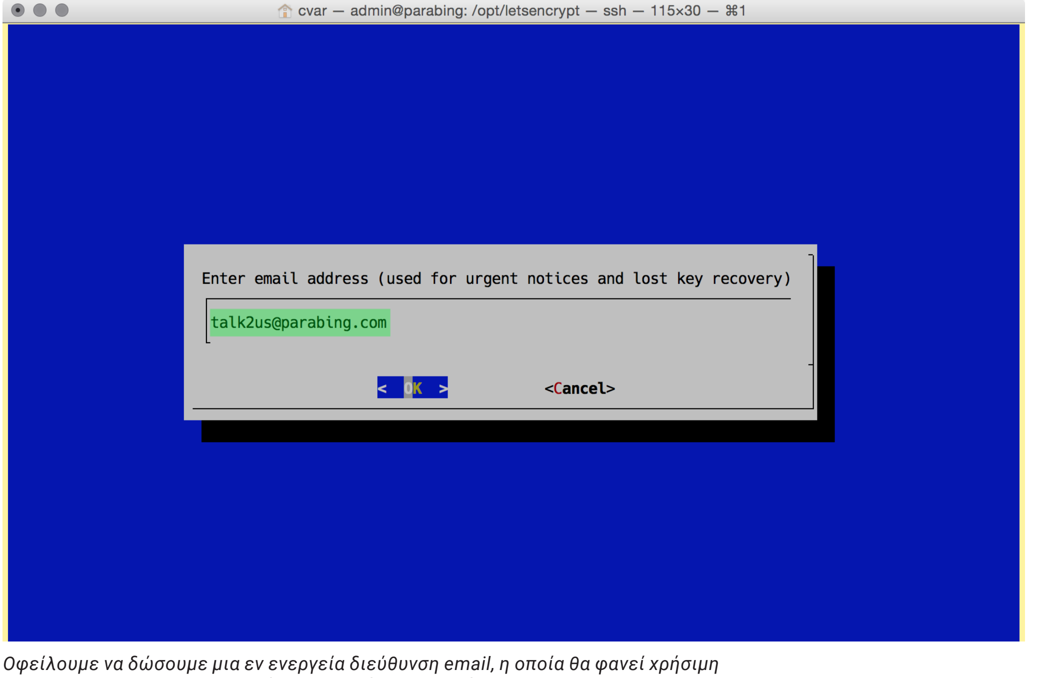
Περιττό να σημειώσουμε ότι ο web server του Let's Encrypt θα παραμείνει ενεργός *μόνο* κατά τη διάρκεια του αρχικού ελέγχου και πολύ σύντομα θα ενεργοποιήσουμε ξανά τον nginx. Μεταβαίνουμε στον τοπικό κατάλογο του letsencrypt

```
admin@parabing:~$ cd /opt/letsencrypt
```

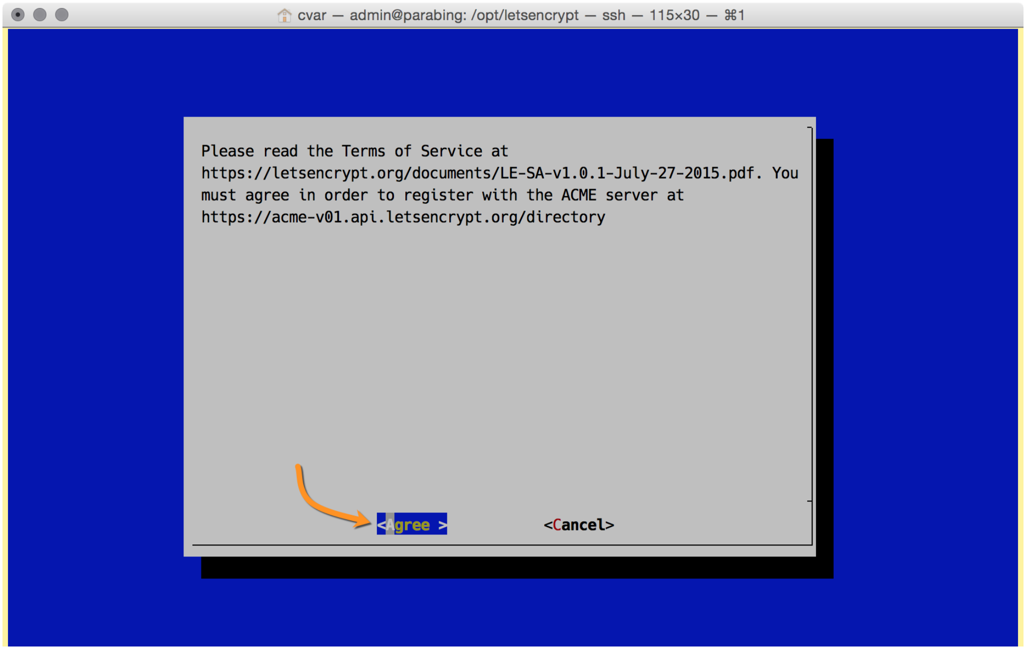
και ζητάμε πιστοποιητικό πληκτρολογώντας:

```
admin@parabing:/opt/letsencrypt$ ./letsencrypt-auto certonly --standalone
```

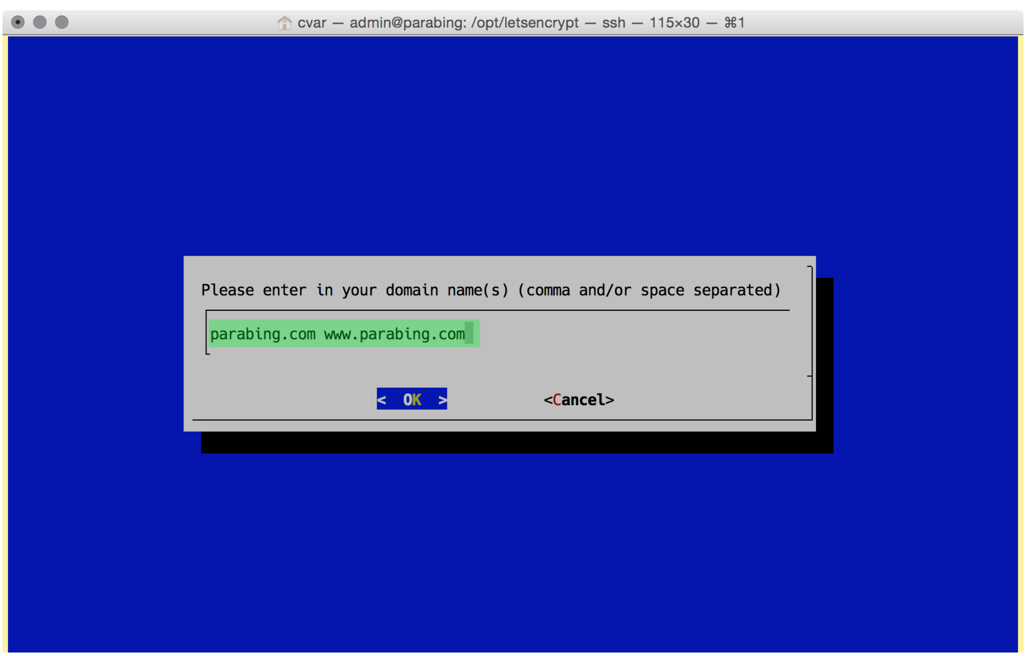
Το letsencrypt θα κατεβάσει μερικά πακέτα που χρειάζεται κι αν έχει περάσει αρκετός χρόνος από την προηγούμενη φορά που χρησιμοποιήσαμε το sudo, τώρα θα μας ζητηθεί password. Δείτε στα τέσσερα ακόλουθα screenshots τη διαδικασία λήψης πιστοποιητικού (διαβάστε και τις περιγραφές).



Οφείλουμε να δώσουμε μια εν ενεργεία διεύθυνση email, η οποία θα φανεί χρήσιμη για επείγουσες ανακοινώσεις ή για την ανάκτηση χαμένου κλειδιού.



Εμείς δεν υπήρχε περίπτωση να διαφωνήσουμε τους όρους υπό τους οποίους μας παρέχεται η υπηρεσία. Για εσάς δεν βάζουμε το χέρι μας στη φωτιά :)



Εδώ πληκτρολογούμε το domain και τα subdomains που θέλουμε να καλύπτει το νέο πιστοποιητικό. Τα ονόματα τα χωρίζουμε μεταξύ τους με ένα κενό.

```
cvar — admin@parabing: /opt/letsencrypt — ssh — 115×30 — ⌘1

IMPORTANT NOTES:
 - If you lose your account credentials, you can recover through
   e-mails sent to talk2us@parabing.com.
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/parabing.com/fullchain.pem. Your cert will
   expire on 2016-03-28. To obtain a new version of the certificate in
   the future, simply run Let's Encrypt again.
 - Your account credentials have been saved in your Let's Encrypt
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Let's
   Encrypt so making regular backups of this folder is ideal.
 - If you like Let's Encrypt, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le

admin@parabing:/opt/letsencrypt$
```

*Αν έχετε ασχοληθεί με τη λήψη πιστοποιητικού από παραδοσιακή Αρχή Πιστοποίησης, μάλλον δεν θα αποφύγετε να εκπλαγείτε –έστω και λίγο– από το πόσο εύκολα μας έδωσε πιστοποιητικό το Let's Encrypt. Παρατηρήστε ότι το νέο μας πιστοποιητικό είναι κάπως βραχύβιο. Μην ανησυχείτε, όμως: σε λίγο θα δείξουμε και πώς επιτυγχάνεται η *αυτόματη* ανανέωσή του.*

Αφού το letsencrypt ολοκληρώσει τη λήψη και την εγκατάσταση του πιστοποιητικού, θα έχουμε τέσσερα νέα αρχεία:

- **cert.pem |** Το πιστοποιητικό για το domain (parabing.com)
- **chain.pem |** To chain sertificate της υπηρεσίας Let's Encrypt
- **fullchain.pem |** Τα αρχεία cert.pem και chain.pem, μαζί
- **privkey.pem |** Το ιδιωτικό κλειδί που αντιστοιχεί στο πιστοποιητικό μας

Τα συγκεκριμένα αρχεία βρίσκονται στον κατάλογο /etc/letsencrypt/archive/parabing.com (γενικά, αποθηκεύονται στο /etc/letsencrypt/archive/domain.tld). Επειδή τα πιστοποιητικά που δίνει το Let's Encrypt ισχύουν για 90 μέρες και μετά λή-

γουν –θα δούμε αργότερα πώς ανανεώνονται αυτόματα–, το letsencrypt δημιουργεί symbolic links προς τα πλέον πρόσφατα αρχεία. Τα links βρίσκονται κάτω από τον κατάλογο /etc/letsencrypt/live/parabing.com (γενικά, κάτω από το τον κατάλογο /etc/letsencrypt/live/domain.tld).

cvar — admin@parabing: /opt/letsencrypt — ssh — 115×30 — ⌘1

```
admin@parabing:/opt/letsencrypt$
admin@parabing:/opt/letsencrypt$ sudo ls -lh /etc/letsencrypt/archive/parabing.com
total 16K
-rw-r--r-- 1 root root 1.8K Dec 29 10:56 cert1.pem
-rw-r--r-- 1 root root 1.7K Dec 29 10:56 chain1.pem
-rw-r--r-- 1 root root 3.5K Dec 29 10:56 fullchain1.pem
-rw-r--r-- 1 root root 1.7K Dec 29 10:56 privkey1.pem
admin@parabing:/opt/letsencrypt$
admin@parabing:/opt/letsencrypt$ sudo ls -lh /etc/letsencrypt/live/parabing.com
total 0
lrwxrwxrwx 1 root root 36 Dec 29 10:56 cert.pem -> ../../archive/parabing.com/cert1.pem
lrwxrwxrwx 1 root root 37 Dec 29 10:56 chain.pem -> ../../archive/parabing.com/chain1.pem
lrwxrwxrwx 1 root root 41 Dec 29 10:56 fullchain.pem -> ../../archive/parabing.com/fullchain1.pem
lrwxrwxrwx 1 root root 39 Dec 29 10:56 privkey.pem -> ../../archive/parabing.com/privkey1.pem
admin@parabing:/opt/letsencrypt$
```

Ιδού τα αρχεία cert.pem, chain.pem, fullchain.pem και privkey.pem, καθώς και τα symbolic links προς τις πιο πρόσφατες εκδοχές τους. Τα πιστοποιητικά που εκδίδει το Let's Encrypt ισχύουν για 90 μέρες, εμείς όμως θα φροντίσουμε για την αυτόματη και έγκαιρη ανανέωση του δικού μας. Τα symbolic links βοηθούν ώστε ο web server να βλέπει πάντα τις πιο πρόσφατες εκδοχές των PEM files, οπότε ανά πάσα στιγμή να χρησιμοποιεί το νεότερο πιστοποιητικό.

Ρύθμιση του web server

Ας ανοίξουμε τώρα το αρχείο ρυθμίσεων του nginx για το site μας. Το εν λόγω αρχείο για το parabing.com είναι το /etc/nginx/sites-available/parabing, ενώ για το deltahacker.gr είναι το /etc/nginx/sites-available/default. Λογικά, κάτω από το /etc/nginx/sites-available θα είναι το αρχείο του δικού σας site.

```
admin@parabing:/opt/letsencrypt$ sudo nano /etc/nginx/sites-available/parabing
```

Μέσα στο server block του αρχείου "σχολιάζουμε" τη γραμμή που υποδεικνύει ότι ο web server ακούει στο port 80. Για παράδειγμα, τη γραμμή

```
listen 80 default_server;
```

εμείς την κάναμε έτσι:

```
#listen 80 default_server;
```

Από κάτω, προσθέτουμε τις ακόλουθες δύο γραμμές:

```
listen 443 ssl;
server_name parabing.com www.parabing.com;
```

Προσέξτε, ώστε στις θέσεις των parabing.com και www.parabing.com να βάλετε το δικό σας domain και subdomain. Ακολουθούν άλλες δύο γραμμές, οι οποίες υποδεικνύουν τις θέσεις των αρχείων fullchain.pem και privkey.pem:

```
ssl_certificate /etc/letsencrypt/live/parabing.com/fullchain.pem;
ssl_certificate_key /etc/letsencrypt/live/parabing.com/privkey.pem;
```

Προκειμένου να γίνονται αποδεκτές μόνον οι πλέον πρόσφατες εκδόσεις του TLS καθώς και οι σύγχρονοι (ισχυροί) αλγόριθμοι κρυπτογράφησης, προσθέτουμε και τις ακόλουθες τρεις γραμμές:

```
ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
ssl_prefer_server_ciphers on;
ssl_ciphers 'EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH';
```

Πολύ ωραία. Είμαστε σχεδόν έτοιμοι. Χρειαζόμαστε μόνο άλλο ένα server block στο τέλος του αρχείου, αλλά *έξω* από το υπάρχον server block:

```
server {
listen 80;
server_name parabing.com;
return 301 https://$host$request_uri;
}
```

Το αποτέλεσμα των παραπάνω γραμμών είναι ότι τα παλιά, δηλαδή τα σκέτα HTTP links, θα συνεχίσουν να δουλεύουν (αφού μεταγράφονται αυτόματα σε HTTPS links). Αποθηκεύουμε τις αλλαγές κι εγκαταλείπουμε τον editor. Εκκινούμε τον nginx με

```
admin@parabing:/opt/letsencrypt$ sudo service nginx start
```

κι από έναν οποιονδήποτε web browser επισκεπτόμαστε το site μας. Όλα πρέπει να δουλεύουν κατά τα αναμενόμενα, μόνο που το site μας θα έχει και το πιστοποιητικό του κι όλες οι συνδέσεις των επισκεπτών θα είναι ασφαλείς.

Ευτυχία.

Αυτόματη ανανέωση πιστοποιητικού

Όπως αναφέραμε, τα πιστοποιητικά που παίρνουμε από το Let's Encrypt ισχύουν για 3 μήνες. Για να μην έχουμε κάποιο ατύχημα και μείνουμε με ληγμένο πιστοποιητικό (στην καλύτερη περίπτωση μιλάμε για *μέγιστο* ρεζιλίκι), καλό είναι να ζητάμε νέο πιστοποιητικό κάθε 2 μήνες και μάλιστα αυτόματα. Προς το παρόν δεν παρέχεται δυνατότητα για αυτόματη ανανέωση πιστοποιητικών, μπορούμε όμως να φτιάξουμε ένα cronjob που σε τακτά χρονικά διαστήματα θα φροντίζει αυτό για την ανανέωση. Επειδή δεν θέλουμε να σταματά ο nginx κατά την ανανέωση, θα καταφύγουμε στο λεγόμενο Webroot plugin του Let's Encrypt. Το plugin χρειάζεται τον κατάλογο .well-know (ναι, με την τελεία), μέσα στο document root (είναι ο φάκελος όπου βρίσκονται τα αρχεία του site μας). Ο nginx πρέπει να είναι σε θέση για εγγραφή εντός του συγκεκριμένου καταλόγου. Προκειμένου να αποφύγουμε θέματα με τα δικαιώματα, ανοίγουμε τώρα το αρχείο ρυθμίσεων του nginx για το site μας (π.χ., /etc/nginx/sites-available/parabing) κι ακριβώς πριν το κλείσιμο του server block με τις οδηγίες περί SSL, προσθέτουμε τις ακόλουθες τρεις γραμμές:

```
location ~ /.well-known {
allow all;
}
```

Αποθηκεύουμε την αλλαγή κι εγκαταλείπουμε τον editor. Για ν' ανανεώσουμε *τώρα* το πιστοποιητικό με χρήση του Webroot plugin, μεταβαίνουμε στον κατάλογο /opt/letsencrypt

```
admin@parabing:~$ cd /opt/letsencrypt
```

και πληκτρολογούμε:

```
admin@parabing:/opt/letsencrypt$ ./letsencrypt-auto certonly -a webroot
--agree-tos --renew-by-default --webroot-path=/srv/www/parabing.com/public_html
-d parabing.com -d www.parabing.com
```

Προσέξτε την τιμή που δώσαμε στην παράμετρο --webroot-path: πρόκειται για την πλήρη διαδρομή του καταλόγου μέσα στον οποίο βρίσκεται το site μας, με άλλα λόγια για το document root. Το παραπάνω παράδειγμα είναι από το parabing.com. Το δε deltahacker.gr βρίσκεται σε άλλο VPS και στον κατάλογο /usr/share/nginx/html, που είναι και το τυπικό document root μετά την εγκατάσταση του nginx σε Ubuntu Server. Έπειτα από την επιτυχή εκτέλεση του εργαλείου letsencrypt-only, επανεκκινούμε και τον nginx:

```
admin@parabing:~$ sudo service nginx reload
```

```
cvar — admin@parabing: /opt/letsencrypt — ssh — 115×30 — ⌘1
admin@parabing:~$
admin@parabing:~$ cd /opt/letsencrypt/
admin@parabing:/opt/letsencrypt$
admin@parabing:/opt/letsencrypt$ ./letsencrypt-auto certonly -a webroot --agree-tos --renew-by-default --webroot-pa
th=/srv/www/parabing.com/public_html -d parabing.com -d www.parabing.com
Updating letsencrypt and virtual environment dependencies......
Requesting root privileges to run with virtualenv: sudo /home/admin/.local/share/letsencrypt/bin/letsencrypt certon
ly -a webroot --agree-tos --renew-by-default --webroot-path=/srv/www/parabing.com/public_html -d parabing.com -d ww
w.parabing.com
[sudo] password for admin:

IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/parabing.com/fullchain.pem. Your cert will
   expire on 2016-03-28. To obtain a new version of the certificate in
   the future, simply run Let's Encrypt again.
 - If you like Let's Encrypt, please consider supporting our work by:

   Donating to ISRG / Let's Encrypt:   https://letsencrypt.org/donate
   Donating to EFF:                    https://eff.org/donate-le

admin@parabing:/opt/letsencrypt$ sudo service nginx reload
 * Reloading nginx configuration nginx                                                                   [ OK ]
admin@parabing:/opt/letsencrypt$
```

Επιτυχής –αλλά χειροκίνητη και κάπως κουραστική– ανανέωση του πιστοποιητικού μας. Σε λίγο θα αυτοματοποιήσουμε τη διαδικασία και θα την ξεχάσουμε.

Από τη στιγμή που είδαμε πώς δουλεύει το Webroot plugin, το επόμενο βήμα είναι να αυτοματοποιήσουμε τη διαδικασία ανανέωσης. Πρώτο μας μέλημα είναι να δημιουργήσουμε ένα configuration file για το letsencrypt-auto. Το προαναφερθέν αρχείο ρυθμίσεων θα ονομάζεται le-renew-webroot.ini, θα κατοικεί μέσα στον κατάλογο /usr/local/etc, ενώ θα βασίζεται στο πρότυπο αρχείο cli.ini από τον κατάλογο /opt/letsencrypt/examples:

```
admin@parabing:/opt/letsencrypt$ sudo cp /opt/letsencrypt/examples/cli.ini /
usr/local/etc/le-renew-webroot.ini
```

Ανοίγουμε το /usr/local/etc/le-renew-webroot.ini με τον editor της προτίμησής μας

```
admin@parabing:/opt/letsencrypt$ sudo nano /usr/local/etc/le-renew-webroot.ini
```

Βγάζουμε τον χαρακτήρα # από τ' αριστερά των γραμμών email, domains και webroot-path, δίνουμε και στις παραμέτρους τις κατάλληλες τιμές για τη διεύθυνση ηλεκτρονικής αλληλογραφίας, το domain/subdomain και το document root της εγκατάστασής μας:

```
email = talk2us@parabing.com
domains = parabing.com, www.parabing.com
webroot-path = /srv/www/parabing.com/public_html
```

Για την αυτόματη ανανέωση είναι εξαιρετικά βολικό το script ονόματι le-renew-webroot που έχει ετοιμάσει ο Mitchell Anicas (https://twitter.com/thisismitch), της Digital Ocean. Φιλοξενείται στο GitHub Gist κι ένα σύντομο URL που οδηγεί στο script είναι το https://bit.ly/le-renew-webroot:

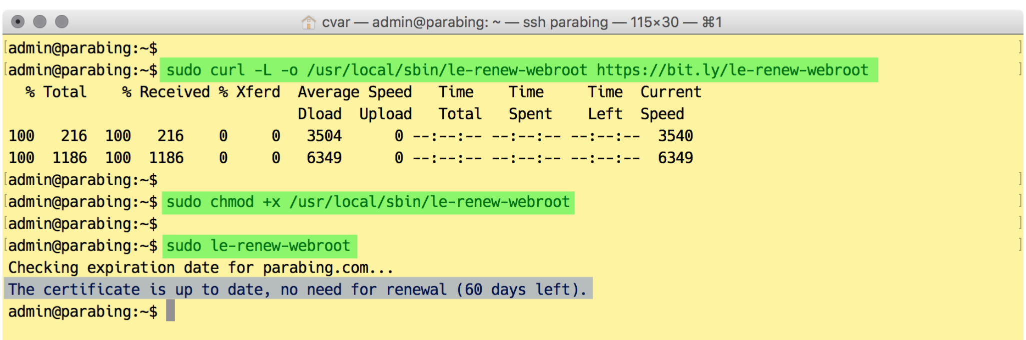
```
admin@parabing:~$ sudo curl -L -o /usr/local/sbin/le-renew-webroot https://bit.
ly/le-renew-webroot
```

Φροντίζουμε ώστε το script να είναι εκτελέσιμο:

```
admin@parabing:~$ sudo chmod +x /usr/local/sbin/le-renew-webroot
```

Αν το τρέξουμε αυτή τη στιγμή δεν θα ανανεώσει το πιστοποιητικό, αλλά θα μας ενημερώσει για πόσον καιρό ισχύει ακόμα:

```
admin@parabing:~$ sudo /usr/local/sbin/le-renew-webroot
Checking expiration date for parabing.com...
The certificate is up to date, no need for renewal (60 days left).
```



Με το βολικό script που έχει ετοιμάσει ο Mitchell Anicas της Digital Ocean θα είναι πολύ εύκολο να ετοιμάσουμε ένα cronjob για την αυτόματη ανανέωση του πιστοποιητικού μας.

Σημείωση. Το script κάνει χρήση του εργαλείου bc, το οποίο στις δικές μας εγκαταστάσεις του Ubuntu Server ήταν ήδη εγκατεστημένο. Αν για οποιονδήποτε λόγο απουσιάζει από τη δική σας, εγκαταστήστε το με ένα "sudo apt-get install bc" (χωρίς τα εισαγωγικά).

Προκειμένου ν' αποφευχθεί το ενδεχόμενο λήξης του πιστοποιητικού, το script προχωρά σε ανανέωση όταν απομένουν λιγότερες από 30 μέρες. Θα ορίσουμε τώρα κι ένα cronjob για την αυτόματη εκτέλεση του script κάθε βδομάδα, οπότε ακόμη κι αν

κάποια φορά και για οποιονδήποτε λόγο η διαδικασία αποτύχει, η πιθανότητα να μείνουμε με ληγμένο πιστοποιητικό θα παραμένει εξαιρετικά μικρή. Για το νέο cronjob δίνουμε

```
admin@parabing:~$ sudo crontab -e
```

και στο τέλος του αρχείου πληκτρολογούμε μια γραμμή σαν την ακόλουθη:

```
0 3 * * 7 /usr/local/sbin/le-renew-webroot >> /var/log/le-renewal.log
```

Η παραπάνω γραμμή ορίζει πως ο έλεγχος για την ανανέωση γίνεται κάθε Κυριακή στις 3 τα ξημερώματα. Χαρείτε το νέο σας site!

```
cvar — admin@parabing: ~ — ssh parabing — 115×30 — ⌘1
  GNU nano 2.2.6                 File: /tmp/crontab.ICYHgc/crontab

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command

0 3 * * 7 /usr/local/sbin/le-renew-webroot >> /var/log/le-renewal.log

^G Get Help    ^O WriteOut    ^R Read File   ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit        ^J Justify     ^W Where Is    ^V Next Page   ^U UnCut Text  ^T To Spell
```

Το cronjob για την αυτόματη ανανέωση του πιστοποιητικού. Ο έλεγχος για το χρόνο που απομένει στο πιστοποιητικό γίνεται κάθε Κυριακή, στις 3 τα ξημερώματα. Αν του απομένουν λιγότερες από 30 μέρες, το πιστοποιητικό ανανεώνεται.


e-card
4000 1234 5678 9010
4000
GOOD THRU
MONTH / YEAR
12 / 19
MasterCard
Cardholder Name
Κάρτα e-coin, που δέχεται Bitcoins: Δουλεύει
όπως όλες οι κανονικές κάρτες Debit,
παντού στον κόσμο και για online αγορές.
Αποκτήστε τη δική σας με κλικ εδώ
και κερδίστε αυτομάτως
25% έκπτωση
κατά την παραγγελία της κάρτας.

Skill: Intermediate
Tags: KVM, domains, lifecycle management, remote access, X11 forwarding, storage pools, virtualization, openSUSE

Το KVM για αληθινούς administrators

Στο τεύχος 049 είδαμε πώς μετατρέπουμε ένα ταπεινό PC σε πανίσχυρο virtualization host, με τη βοήθεια του openSUSE και του KVM. Το GNOME desktop που επιλέξαμε για το openSUSE αφενός διευκόλυνε τη γνωριμία μας με το KVM, αφετέρου τον συγκεκριμένο host μερικές φορές θέλουμε να τον χρησιμοποιούμε κι ως απλό σύστημα desktop. Στο παρόν άρθρο δείχνουμε πώς δημιουργούμε και διαχειριζόμαστε KVM VMs αποκλειστικά από τη γραμμή εντολών.

του Χρήστου Βαρελά

Ειδικά όσοι φιλοδοξούν ν' ασχοληθούν *σοβαρά* με τη διαχείριση σύγχρονων συστημάτων και δικτύων, πιστεύουμε ότι αξίζει να δώσουν προσοχή και φυσικά να πειραματιστούν στην πράξη. Πριν ξεκινήσουμε, ας σημειώσουμε ότι αν δεν έχετε ακόμη διαβάσει τα σχετικά άρθρα στο τεύχος 049, καλό θα ήταν να το πράξετε τώρα (βλ. http://deltahacker.gr/?p=15388 και http://deltahacker.gr/?p=15425).

Για τη συνέχεια δουλεύουμε είτε τοπικά, από μια κονσόλα του virtualization host, είτε απομακρυσμένα, μέσω SSH. Το username του δικού μας χρήστη είναι cvar, ενώ το hostname του μηχανήματος είναι thehost. Οι εντολές που δίνουμε απαιτούν δικαιώματα διαχειριστή συστήματος, γι' αυτό και κάνουμε χρήση του sudo.

Βασική διαχείριση υπαρχόντων KVM VMs

Θα καταφύγουμε στο εργαλείο virsh. Τις υπάρχουσες εικονικές μηχανές, ασχέτως αν είναι ενεργές ή όχι, τις βλέπουμε πληκτρολογώντας

```
cvar@thehost:~> sudo virsh list --all
 Id    Name                           State
----------------------------------------------------
 1     Plex                           running
 2     DbxNode                        running
 3     OpenBSD                        running
 -     FreeBSD                        shut off
cvar@thehost:~>
```

Ας εξετάσουμε τις στήλες του πίνακα. Στην "Id" βλέπουμε έναν μοναδικό ακέραιο αριθμό, ο οποίος αντιστοιχίζεται σε κάθε ενεργή μηχανή. Στη στήλη "Name" υπάρχουν τα ονόματα των μηχανών, όπως ορίστηκαν κατά τη δημιουργία τους. Στη δε στήλη

"State" αναφέρεται η κατάσταση κάθε μηχανής. Στο παράδειγμά μας οι μηχανές Plex, DbxNode και OpenBSD είναι ενεργές (running), ενώ η FreeBSD είναι ανενεργή (shut off). Στο virsh δώσαμε την εντολή list και την παράμετρο --all. Παραλείποντας την τελευταία, παίρνουμε μια λίστα με τις ενεργές μηχανές και μόνον αυτές. Προκειμένου να ξεκινήσουμε μια ανενεργή μηχανή, αρκεί να γράψουμε κάτι σαν

```
cvar@thehost:~> sudo virsh start FreeBSD
Domain FreeBSD started
cvar@thehost:~>
```

Μετά την εντολή start απλά παραθέσαμε το όνομα της εικονικής μηχανής που αποφασίσαμε να ενεργοποιήσουμε. Παρατηρήστε το μήνυμα που μας επεστράφη: τα virtual machines ονομάζονται και domains – και η συγκεκριμένη ορολογία προέρχεται από τον κόσμο του Xen. Προκειμένου να κλείσουμε ομαλά ένα VM, όπου με το "ομαλά" εννοούμε όπως θα κλείναμε έναν φυσικό υπολογιστή, χωρίς να τον βγάλουμε από την πρίζα, πληκτρολογούμε κάτι σαν

```
cvar@thehost:~> sudo virsh shutdown FreeBSD
Domain FreeBSD is being shutdown
cvar@thehost:~>
```

Στη θέση του ονόματος (FreeBSD) θα μπορούσαμε να είχαμε βάλει το Id. Όπως είδαμε πριν από λίγο, τα Ids και τα ονόματα των μηχανών παρατίθενται στην έξοδο της εντολής "virsh list". Για την επανεκκίνηση (reboot) μιας μηχανής, δίνουμε

```
cvar@thehost:~> sudo virsh reboot OpenBSD
Domain OpenBSD is being rebooted
cvar@thehost:~>
```

Είναι πιθανό κάποια μηχανή να μην ανταποκρίνεται σε εντολές για reboot ή shutdown. Συνήθεις λόγοι είναι η ελλιπής –ή η καθόλου– υποστήριξη του προτύπου ACPI, όπως επίσης και οι ρυθμίσεις του guest OS. Κατά περιπτώσεις, όπως, π.χ., μέσα από κάποιο script, θα θέλουμε να κλείνουμε ακόμη κι αυτές τις μηχανές, οπότε η εντολή που θα περνάμε στο virsh θα είναι η destroy:

```
cvar@thehost:~> sudo virsh destroy OpenBSD
Domain OpenBSD destroyed
cvar@thehost:~>
```

Όσα συζητήσαμε έως τώρα για το πώς ξεκινάμε, σταματάμε κι επανεκκινούμε εικονικές μηχανές, αφορούν στο λεγόμενο lifecycle management των VMs. Σίγουρα δεν είναι από τις πιο δύσκολες εργασίες, όμως σε διάφορα περιβάλλοντα είναι

ιδιαίτερα σημαντικές κι ενίοτε εκτελούνται αυτοματοποιημένα, από κατάλληλα scripts. Παρεμπιπτόντως, ειδικά για το αυτόματο boot ενός VM κατά την εκκίνηση του host OS αρκεί να πληκτρολογήσουμε (μία φορά) το ακόλουθο:

```
cvar@thehost:~> sudo virsh autostart FreeBSD
Domain FreeBSD marked as autostarted
cvar@thehost:~>
```

Αργότερα, αν για κάποιο λόγο αποφασίσουμε ότι ένα VM δεν πρέπει να bootάρει αυτόματα κατά την εκκίνηση του host OS, αρκεί να δώσουμε κάτι σαν

```
cvar@thehost:~> sudo virsh autostart --disable FreeBSD
Domain FreeBSD unmarked as autostarted
cvar@thehost:~>
```

Πρόσβαση σε κονσόλα KVM VM, απομακρυσμένα

Ο προφανής τρόπος είναι να καταφύγουμε σε κάποια μέθοδο που υποστηρίζει το guest OS του VM που μας ενδιαφέρει. Αν, π.χ., έχουμε ένα Unix-like guest OS, τότε πιθανώς να συνδεθούμε σ' αυτό μέσω SSH ή VNC. Αν πάλι μιλάμε για Windows guest, τότε μια δυνατότητα σύνδεσης παρέχεται από το Remote Desktop Protocol. Τι γίνεται όμως αν το guest OS δεν έχει καν εγκατασταθεί ή έχει εγκατασταθεί αλλά *δεν* παρέχει κάποια μέθοδο απομακρυσμένης σύνδεσης; Σε μια τέτοια περίπτωση το ιδανικό είναι να έχουμε μπροστά μας την κονσόλα του VM, ασχέτως αν αυτή είναι text ή graphics-based, όπως έχουμε μπροστά μας την οθόνη ενός φυσικού υπολογιστή. Αυτό θα το πετυχαίναμε αν, π.χ., τρέχαμε τον Virtual Machine Manager στο απομακρυσμένο host computer, δηλαδή στον virtualization host, αλλά το λεγόμενο display του προγράμματος γινόταν στην οθόνη του υπολογιστή από τον οποίο τώρα εργαζόμαστε. Θα υποψιάζεστε, βεβαίως, πως ό,τι περιγράψαμε είναι πέρα για πέρα εφικτό. Και πράγματι είναι, αρκεί να ισχύουν οι τρεις ακόλουθες προϋποθέσεις:

Π1. Στον virtualization host τρέχει SSH server, ο οποίος επιτρέπει το λεγόμενο X11 forwarding.

Π2. Ο απλός χρήστης του virtualization host, στο λογαριασμό του οποίου κάνουμε SSH login, ανήκει στο group ονόματι libvirt.

Π3. Στον υπολογιστή από τον οποίο εργαζόμαστε υπάρχει πρόγραμμα για συνδέσεις SSH, αλλά και κάποιος X server.

Ας δούμε τις Π1, Π2 και Π3 αναλυτικά.

Π1. Ο virtualization host με το openSUSE τρέχει ήδη SSH server, ο οποίος δέχεται συνδέσεις από οποιονδήποτε client. Για λόγους ασφαλείας *δεν* επιτρέπει SSH logins στο λογαριασμό του χρήστη root. Επίσης, για τους απλούς χρήστες

επιτρέπει *μόνο* passwordless logins, με βάση τα αντίστοιχα δημόσια κλειδιά SSH (http://deltahacker.gr/?p=13357). Τέλος, ο SSH server του openSUSE μας εξ ορισμού επιτρέπει το X11 forwarding – όπως άλλωστε συμβαίνει και με πολλές άλλες διανομές Linux.

Π2. Ο απλός μας χρήστης θα πρέπει να ανήκει στο group ονόματι libvirt, διαφορετικά δεν θα μπορεί να διαχειρίζεται εικονικές μηχανές. Από τη στιγμή βέβαια που κάνουμε SSH login στο λογαριασμό του, θα μπορούσε κάποιος να προτείνει κι ένα sudo su ώστε να περάσουμε στο λογαριασμό του root. Το θέμα είναι ότι κατ' αυτόν τον τρόπο περιπλέκονται τα πράγματα με την ανακατεύθυνση του display των εφαρμογών γραφικών, οπότε καλύτερα να μην ακολουθήσουμε τη συγκεκριμένη οδό. Τώρα, ένας τρόπος για να βάλουμε τον απλό μας χρήστη στο group libvirt, είναι από τη γραμμή εντολών και με τη βοήθεια του εργαλείου usermod:

```
cvar@thehost:~> sudo usermod -a -G libvirt cvar
```

Η προσθήκη του cvar στο group libvirt δεν έχει ληφθεί ακόμη υπόψη, όπως φανερώνει και η έξοδος του εργαλείου groups:

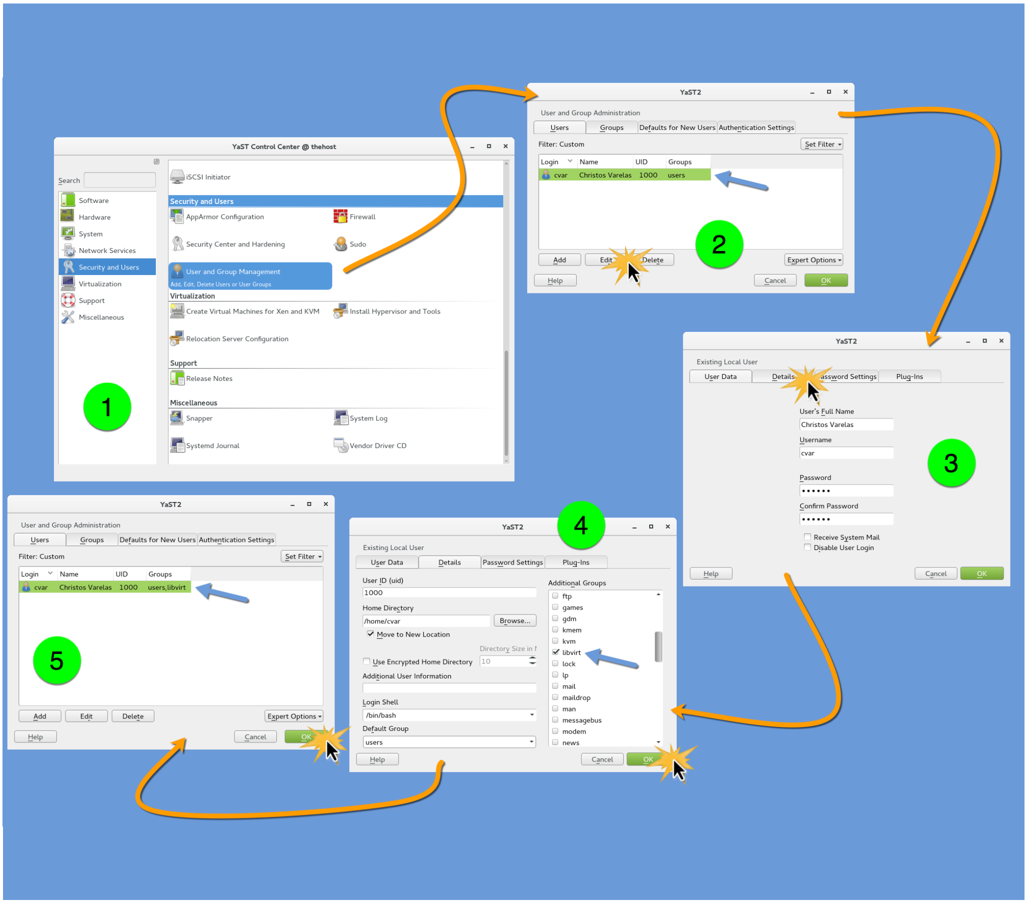
```
cvar@thehost:~> groups
users
```

Στο σημείο αυτό μπορούμε να κάνουμε ένα logout και ξανά login και θα δούμε ότι ο χρήστης μας όντως ανήκει στο group libvirt. Εναλλακτικά, καταφεύγουμε στο newgrp:

```
cvar@thehost:~> newgrp libvirt
cvar@thehost:~> groups
libvirt users
cvar@thehost:~>
```

Όλα καλά.

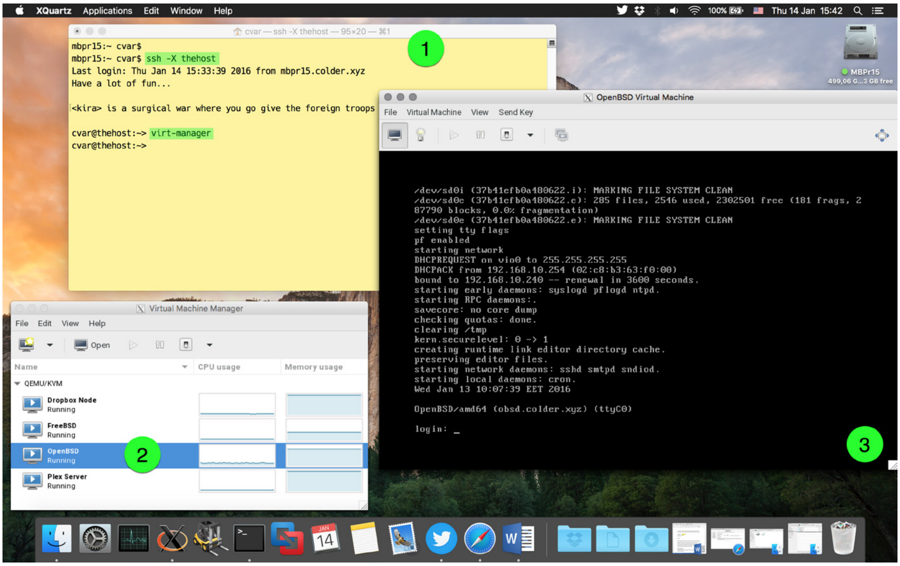
Π3. Αν δουλεύουμε από σύστημα **Linux** ή ***BSD**, στο οποίο είναι εγκατεστημένος X server, αρκεί να συνδεθούμε στον virtualization host με το ssh δίνοντάς του την παράμετρο -X (κεφαλαίο). Πληκτρολογώντας στο τερματικό το όνομα μιας απομακρυσμένης εφαρμογής γραφικών, το display της θα ανακατευθύνεται στην οθόνη του υπολογιστή μας. Αν πάλι δουλεύουμε από **OS X**, κατά πάσα πιθανότητα θα χρειαστεί να εγκαταστήσουμε πρώτα το πακέτο XQuartz (http://www.xquartz.org). Από εκεί και μετά, όποτε θέλουμε να τρέξουμε μια απομακρυσμένη εφαρμογή γραφικών θα ανοίγουμε ένα Terminal, θα συνδεόμαστε στο openSUSE με το ssh και την παράμετρο -X, και φυσικά θα πληκτρολογούμε το όνομα της εφαρμογής. Σημειώστε ότι δεν χρειάζεται να ξεκινάμε μόνοι μας το XQuartz, αφού κάθε φορά που το OS X το χρειάζεται το ενεργοποιεί αυτόματα. Τελευταία αφήσαμε την περίπτωση



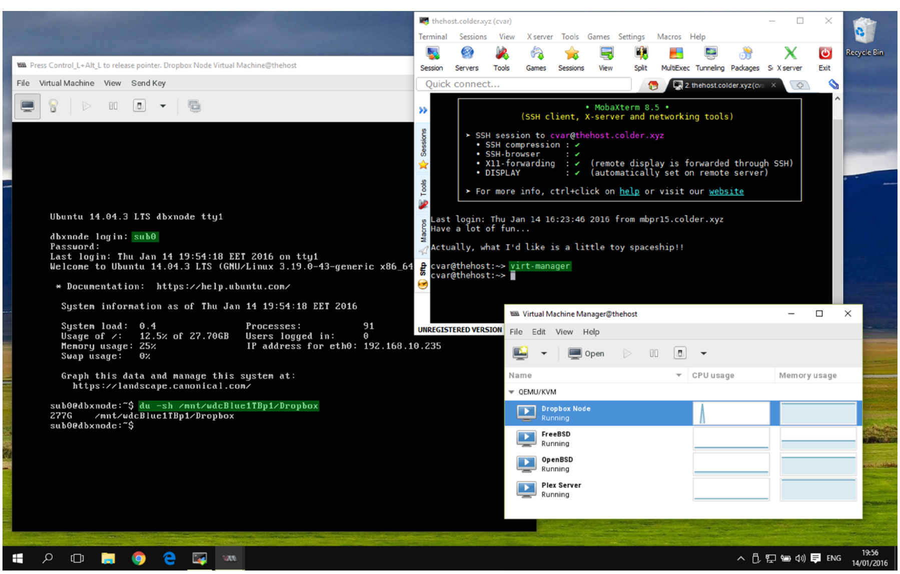
Με το πολυεργαλείο διαχείρισης συστήματος YaST του openSUSE μπορούμε να κάνουμε σχεδόν τα πάντα, ακόμη και τις ταπεινές εργασίες όπως είναι η προσθήκη ενός χρήστη σ' ένα νέο group. Φανταζόμαστε, πάντως, πως θα συμφωνήσετε κι εσείς ότι ορισμένες τέτοιες "ταπεινές εργασίες" γίνονται πολύ πιο γρήγορα από τη γραμμή εντολών.

των **Windows**. Εξαιρετικά δημοφιλής SSH client για τα Windows, με δυνατότητες για X11 forwarding, είναι το PuTTY. Πρέπει βέβαια να εγκαταστήσουμε και κάποιον X server, όπως, π.χ., είναι το Xming. Μια εναλλακτική πρόταση είναι το MobaXterm (http://mobaxterm.mobatek.net), το οποίο έχει *και* SSH client *και* X server.

Σε κάθε περίπτωση, αφού συνδεθούμε μέσω SSH (και την παράμετρο -X ή το ισοδύναμό της) στο λογαριασμό του χρήστη μας στον virtualization host, πληκτρολογούμε απλά "virt-manager" (χωρίς τα εισαγωγικά) και πατάμε [Enter]. Στην οθόνη μας θα δούμε το παράθυρο του Virtual Machine Manager και με διπλό κλικ στη γραμμή του επιθυμητού VM θα ανοίξει η κονσόλα του.



Από το laptop μας με OS X El Capitan ανοίγουμε ένα τερματικό, συνδεόμαστε με το ssh στον virtualization host (προσέξτε την παράμετρο -X) και πληκτρολογούμε "virt-manager", χωρίς τα εισαγωγικά (1). Το OS X σηκώνει το XQuartz και μερικά πικοδευτερόλεπτα αργότερα, τα οποία για ένα συγκεκριμένο επίπεδο ύπαρξης συγγενεύουν με την αιωνιότητα, στην ωραία μας οθόνη βλέπουμε το παράθυρο του Virtual Machine Manager, ο οποίος φυσικά τρέχει στο PC με το openSUSE. Με κλικ πάνω στο επιθυμητό KVM VM (2), στην οθόνη έχουμε και την κονσόλα του (3). Η ρουτίνα που μόλις περιγράψαμε είναι ιδιαίτερα βολική, π.χ., όταν σε κάποιο VM δεν έχει ακόμα εγκατασταθεί το guest OS και ταυτόχρονα αδυνατούμε ή έστω δεν μας βολεύει να πάμε να καθήσουμε μπροστά από τον physical host, ώστε να την προχωρήσουμε.



Το MobaXterm είναι μια ενδιαφέρουσα πρόταση για τους χρήστες των Windows που αναζητούν SSH client και X server, ώστε να είναι σε θέση να τρέχουν απομακρυσμένες εφαρμογές Unix και να έχουν το display τους τοπικά.

Δημιουργία KVM VMs

Μέχρι εδώ συζητήσαμε για το lifecycle management των VMs. Δείξαμε επίσης κι έναν ασφαλή τρόπο για την "ανακατεύθυνση" της κονσόλας ενός απομακρυσμένου VM, στην οθόνη του τοπικού μας υπολογιστή. Έχει φτάσει ώρα να ασχοληθούμε και με τη δημιουργία νέων KVM VMs – εννοείται από τη γραμμή εντολών.

Πριν φτιάξουμε ένα VM οφείλουμε ν' αποφασίσουμε για το είδος του δίσκου του, βεβαίως και να τον δημιουργήσουμε. Μια ενδιαφέρουσα δυνατότητα για το δίσκο είναι ν' αποτελεί logical volume (LV) ενός προϋπάρχοντος volume group (VG). Γενικά για το Logical Volume Management (LVM) μπορείτε να διαβάσετε στα http://deltahacker.gr/?p=1236 και http://deltahacker.gr/?p=1291. Η συγκεκριμένη προσέγγιση παρέχει ευελιξία και ταχύτητα. Από την άλλη, ένα τέτοιο setup ίσως είναι περιττά περίπλοκο για απλά σενάρια χρήσης (όπως είναι τα δικά μας). Επιπρόσθετα, τους δίσκους που κατοικούν σε LVs δεν μπορούμε να τους διαχειριζόμαστε εύκολα με κλασικά εργαλεία όπως, π.χ., είναι το scp. Για τη συνέχεια, λοιπόν, ας εστιάσουμε στους εικονικούς δίσκους οι οποίοι ουσιαστικά αποτελούν ένα απλό αρχείο στο filesystem του host OS (εν προκειμένω του openSUSE).

Δύο είναι τα βασικά formats των αρχείων-εικονικών δίσκων: RAW και QCOW2. Το format **RAW** είναι το πιο απλό και γρήγορο, αλλά και με τις λιγότερες δυνατότητες. Σημειώστε ότι ένα αρχείο RAW επιτρέπεται να είναι sparse, που σημαίνει ότι ο συνολικός του χώρος δεν κατανέμεται κατά τη δημιουργία του αλλά το αρχείο μεγαλώνει δυναμικά, καθώς το guest OS του αντίστοιχου VM αμέριμνα γράφει δεδομένα στο δίσκο. Όπως πιθανώς θα φαντάζεστε, τα RAW files που είναι sparse παρέχουν χαμηλότερη ταχύτητα προσπέλασης σε σύγκριση με τα RAW files που *δεν* είναι sparse (όλη η έκτασή τους κατανέμεται στο filesystem του host κατά τη δημιουργία τους). Τα αρχεία εξάλλου format **QCOW2** μεγαλώνουν κι αυτά δυναμικά, ενώ προσφέρουν και κάποιες έξτρα δυνατότητες όπως, π.χ., το snapshotting. Οι δυνατότητες όμως έχουν και το τίμημά τους, το οποίο δεν είναι άλλο από τη συγκριτικά χαμηλότερη ταχύτητα πρόσβασης. Τελικά, για τις δικές μας απλές ανάγκες προτιμάμε τα sparse RAW files: συνδυάζουν ταχύτητα πρόσβασης και οικονομία χώρου, ενώ και η μεταφορά τους σε άλλο δίσκο ή host computer γίνεται πανεύκολα.

Για τη δημιουργία ενός sparse RAW file, το οποίο σε πολύ λίγο θα αντιστοιχίσουμε στη νέα μας εικονική μηχανή, πληκτρολογούμε κάτι τέτοιο:

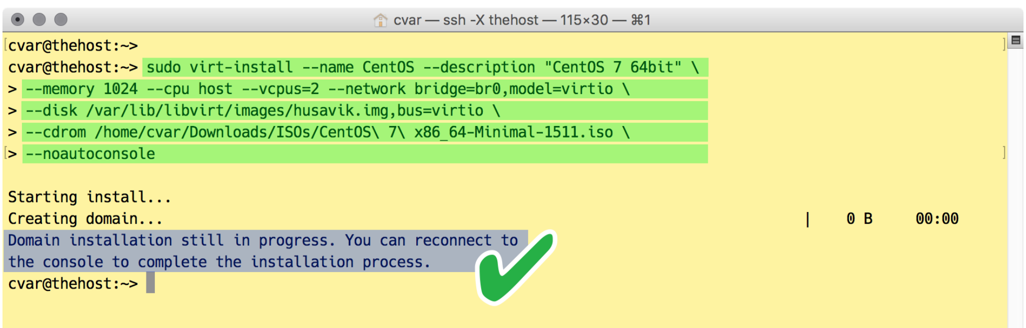
```
cvar@thehost:~> sudo truncate --size=24576M /var/lib/libvirt/images/husavik.img
```

Το νέο αρχείο ονομάσαμε husavik.img και το αποθηκεύσαμε κάτω από τον κατάλογο /var/lib/libvirt/images. Δεν μπορούμε να το καθυστερήσουμε άλλο: έφτασε η στιγμή της δημιουργίας της πρώτης μας εικονικής μηχανής από τη γραμμή εντολών, η οποία ως δίσκο της θα έχει το husavik.img. Καταφεύγουμε στο εργαλείο virt-install:

```
cvar@thehost:~> sudo virt-install --name CentOS --description "CentOS 7 64bit" \
```

```
--memory 1024 --cpu host --vcpus=2 --network bridge=br0,model=virtio \
--disk /var/lib/libvirt/images/husavik.img,bus=virtio \
--cdrom /home/cvar/Downloads/ISOs/CentOS\ 7\ x86_64-Minimal-1511.iso \
--noautoconsole
```

Οι παράμετροι που περάσαμε στο virt-install δεν είναι λίγες, γι' αυτό κι ολόκληρη την εντολή τη "σπάσαμε" σε περισσότερες από μία γραμμές. Όποτε θέλαμε ν' αλλάξουμε γραμή αφήναμε ένα κενό, μετά βάζαμε τον χαρακτήρα "\" (χωρίς τα εισαγωγικά) και πατούσαμε [Enter]. Στην τελευταία γραμμή πατήσαμε μόνο το [Enter] και το BASH έλαβε υπόψη όλα όσα γράψαμε μετά το prompt, σαν να είχαν πληκτρολογηθεί σε μία μόνο γραμμή.



Απομακρυσμένα, κι εργαζόμενοι από το ωραίο τερματικό μας, μόλις δημιουργήσαμε μια νέα εικονική μηχανή. Τα βασικά της υποσυστήματα έχουν ως ακολούθως: 1GB RAM, dual-core CPU με χαρακτηριστικά παρόμοια μ' εκείνα του host computer, δικτύωση τύπου bridged, ώστε το VM να συμμετέχει κανονικά στο τοπικό δίκτυο όπως κάθε άλλος εικονικός ή φυσικός υπολογιστής, σκληρός δίσκος μεγέθους 24GB. Για το δίσκο αλλά και τη δικτύωση χρησιμοποιείται ο VirtIO driver, γεγονός που σημαίνει βελτιωμένες επιδόσεις. Το όνομα της μηχανής είναι CentOS και η εγκατάσταση θα γίνει από το ISO image ονόματι CentOS 7 x86_64-Minimal-1511.iso, μέσα στον κατάλογο /home/cvar/Downloads/ISOs του host. Χάρη εξάλλου στην τελευταία παράμετρο, την --noautoconsole, δεν επιχειρείται το άνοιγμα κάποιας κονσόλας προς το VM. Αυτό θα το κάνουμε εμείς, όποτε κι απ' όπου μας βολεύει. Στο μεταξύ το VM θα έχει ενεργοποιηθεί, θα έχει bootάρει από το εικονικό CDROM και θα περιμένει υπομονετικά ν' ασχοληθούμε με την εγκατάσταση του guest OS.

Ό,τι πληκτρολογήσαμε δείχνει αρκετά περίπλοκο και κάποιοι θα σκεφτούν ότι όλη αυτή η φασαρία δεν αξίζει τον κόπο. Λέμε ότι οι συγκεκριμένοι "κάποιοι" έχουν λάθος. Αν πιστεύετε ότι το λάθος είναι δικό μας, σας προ(σ)καλούμε σε μια πιο προσεκτική εξέταση των παραμέτρων που δώσαμε στο virt-install.

--name CentOS

Το νέο μας VM το ονομάζουμε CentOS. Είναι γιατί πρόκειται να του εγκαταστήσουμε το CentOS. Αν δεν ήταν να του εγκαταστήσουμε το CentOS, θα το ονομάζαμε κάπως αλλιώς. Να, ένα όνομα που τώρα μας έρχεται στο νου είναι το Keflavíkurflugvöllur. Σημειώστε ότι το όνομα κάθε KVM VM (domain) πρέπει να είναι μοναδικό και να μην περιλαμβάνει λευκούς ή ειδικούς χαρακτήρες.

--description "CentOS 7 64bit"

Δεν είναι υποχρεωτικό να δώσουμε περιγραφή για το VM, όμως είμαστε από εκείνους που πιστεύουν ακράδαντα στις περιγραφές – ειδικά όταν αφορούν σε VMs. Ναι, τέτοιοι είμαστε.

--memory 1024

Το νέο μας VM θα έχει μνήμη μεγέθους 1GB. Είναι ό,τι πρέπει για ένα server-oriented μηχάνημα με Linux. Εννοείται πως αν διαπιστώσουμε ότι το 1 gigabyte είναι λίγο ή πολύ, ανά πάσα στιγμή θα μπορούμε να δώσουμε στο VM περισσότερη μνήμη ή να του ελαττώσουμε την υπάρχουσα.

--cpu host

Προαιρετική παράμετρος. Η CPU του εικονικού υπολογιστή θα έχει χαρακτηριστικά παρόμοια μ' εκείνα του φυσικού υπολογιστή. Αν σκοπεύουμε να μετακινήσουμε το VM σε host με διαφορετικό επεξεργαστή από εκείνον του τωρινού host, ίσως πρέπει να παραλείψουμε τη συγκεκριμένη παράμετρο.

--vcpus=2

Το VM μας θα έχει δύο εικονικούς επεξεργαστές. Ακριβέστερα: Λόγω της προηγουμένης παραμέτρου, θα έχει έναν επεξεργαστή με δύο πυρήνες.

--network bridge=br0,model=virtio

Για την εικονική μηχανή ζητάμε bridged networking. Μπορούμε να φανταζόμαστε ότι το VM θα έχει μια εικονική κάρτα Ethernet που θα παίρνει IP από τον DHCP server του φυσικού τοπικού δικτύου. Η παράμετρος virtio για το model σημαίνει αυξημένες επιδόσεις στη δικτυακή επικοινωνία μεταξύ διαφορετικών VMs, κάτι που το έχουμε διαπιστώσει και στην πράξη με απλές μετρήσεις (και με τη βοήθεια του iperf).

--disk /var/lib/libvirt/images/husavik.img,bus=virtio

Ο δίσκος του VM είναι το sparse RAW file ονόματι husavik.img, κάτω από το /var/lib/libvirt/images. Όπως και πριν, με την κάρτα Ethernet, η παράμετρος virtio για το bus σημαίνει αυξημένες επιδόσεις.

--cdrom /home/cvar/Downloads/ISOs/CentOS\ 7\ x86_64-Minimal-1511.iso

Στον εικονικό οδηγό CD της μηχανής είναι σαν να έχουμε βάλει ένα δισκάκι με το CentOS (έκδοση 7, εκδοχή 64bit). Εξ ορισμού, κατά την πρώτη ενεργοποίηση της μηχανής το boot γίνεται από τον εικονικό οδηγό CD. Πρακτικά, αυτό σημαίνει ότι κατά την πρώτη ενεργοποίηση θα ξεκινήσει η εγκατάσταση του guest OS.

--noautoconsole

Χάρη σ' αυτή την παράμετρο, αμέσως μετά τη δημιουργία της μηχανής, ναι μεν αυτή θα ενεργοποιηθεί, αλλά δεν θα επιχειρηθεί το άνοιγμα κάποιας κονσόλας. Στην περίπτωσή μας, εξαιτίας **α)** της απομακρυσμένης σύνδεσης, **β)** του sudo και **γ)** των προεπιλογών ασφαλείας του openSUSE, το άνοιγμα της κονσόλας ούτως ή άλλως θα αποτύγχανε. Σε κάθε περίπτωση το VM ενεργοποιείται, bootάρει από το CD (ISO image) εγκατάστασης του CentOS και μας περιμένει ώστε, με κάποιον τρόπο,

να προχωρήσουμε με την εγκατάσταση του OS. Όπως είδαμε στην προηγούμενη ενότητα, ο χρήστης cvar είναι στο group libvirt κι αυτό σημαίνει ότι μπορεί να τρέχει τον Virtual Machine Manager χωρίς sudo, ακόμη κι απομακρυσμένα, και ν' ανοίγει κονσόλες προς ένα ή περισσότερα VMs.

Θεσπέσια.

Τροποποίηση υπαρχόντων KVM VMs

Κατά τα αναμενόμενα, για την τροποποίηση των ιδιοτήτων ενός KVM VM δεν είναι ανάγκη να καταφύγουμε στον Virtual Machine Manager. Κάλλιστα μπορούμε να εργαστούμε από τη γραμμή εντολών του τερματικού μας. Ας υποθέσουμε, π.χ., ότι θέλουμε να τροποποιήσουμε την ποσότητας της μνήμης του VM με όνομα CentOS. Ξεκινάμε με ένα shutdown της μηχανής:

```
cvar@thehost:~> sudo virsh shutdown CentOS
Domain CentOS is being shutdown
cvar@thehost:~>
```

Το VM ενδέχεται να μην κλείσει ακαριαία, οπότε πριν προχωρήσουμε ας βεβαιωθούμε για την κατάστασή του:

```
cvar@thehost:~> sudo virsh list --all
 Id    Name                           State
----------------------------------------------------
 1     Plex                           running
 25    DbxNode                        running
 -     CentOS                         shut off
 -     FreeBSD                        shut off
 -     OpenBSD                        shut off
cvar@thehost:~>
```

Ωραία, έχει κλείσει. Τα χαρακτηριστικά των VMs αποθηκεύονται σε αρχεία XML (ένα XML για κάθε VM), τα οποία κατοικούν κάτω από τον κατάλογο /etc/libvirt/qemu. Τα εν λόγω αρχεία δεν τα ανοίγουμε απευθείας, ρίξτε όμως μια ματιά στον σχετικό κατάλογο:

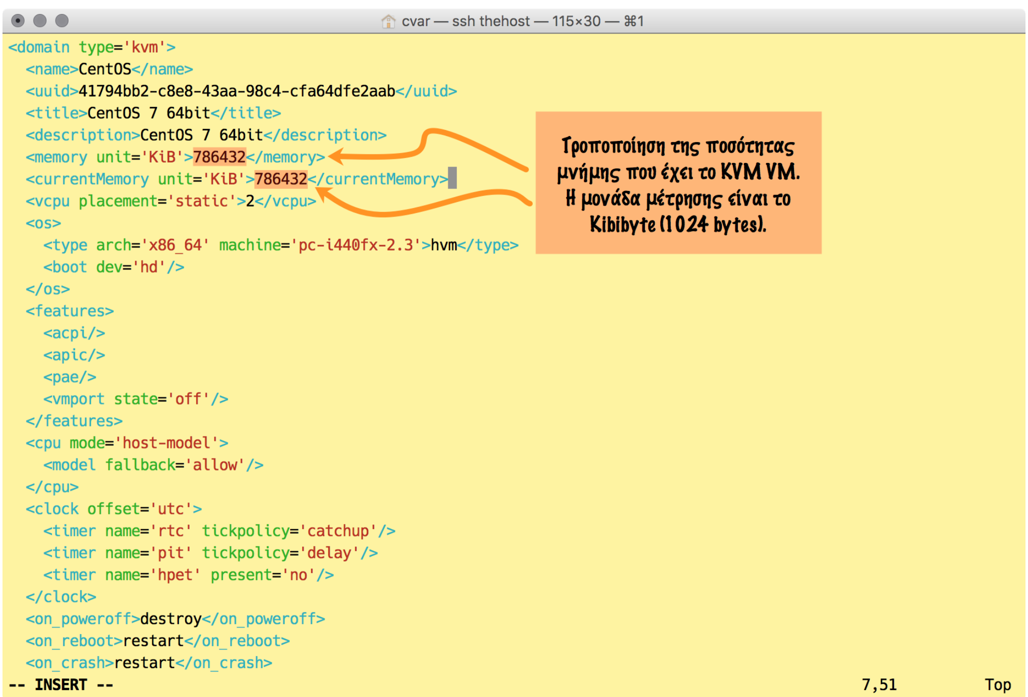
```
cvar@thehost:~> sudo ls -lh /etc/libvirt/qemu
total 32K
drwxr-xr-x 1 root root   38 Jan 17 08:04 autostart
-rw------- 1 root root 4.4K Jan 16 21:59 CentOS.xml
-rw------- 1 root root 4.1K Jan 16 21:48 DbxNode.xml
```

```
-rw------- 1 root root 4.0K Dec 24 21:24 FreeBSD.xml
drwx------ 1 root root   40 Dec  9 19:23 networks
-rw------- 1 root root 3.9K Dec 12 18:06 OpenBSD.xml
-rw------- 1 root root 4.8K Jan  8 22:31 Plex.xml
cvar@thehost:~>
```

Θέλουμε να τροποποιήσουμε το CentOS.xml και για να το πετύχουμε γράφουμε

```
cvar@thehost:~> sudo virsh edit CentOS
```

(χωρίς την κατάληξη .xml). Ανοίγει τότε, στον προεπιλεγμένο text editor, το αρχείο XML της μηχανής. Ρίχνοντας μια ματιά στο αρχείο είναι εύκολο να διακρίνουμε τις βασικές παραμέτρους του VM. Ειδικά για τη μνήμη RAM, εστιάζουμε την προσοχή μας πάνω πάνω στο αρχείο και συγκεκριμένα στα δύο μπλοκ ονόματι memory unit και currentMemory unit.



```
<domain type='kvm'>
  <name>CentOS</name>
  <uuid>41794bb2-c8e8-43aa-98c4-cfa64dfe2aab</uuid>
  <title>CentOS 7 64bit</title>
  <description>CentOS 7 64bit</description>
  <memory unit='KiB'>786432</memory>
  <currentMemory unit='KiB'>786432</currentMemory>
  <vcpu placement='static'>2</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-2.3'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
    <vmport state='off'/>
  </features>
  <cpu mode='host-model'>
    <model fallback='allow'/>
  </cpu>
  <clock offset='utc'>
    <timer name='rtc' tickpolicy='catchup'/>
    <timer name='pit' tickpolicy='delay'/>
    <timer name='hpet' present='no'/>
  </clock>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
-- INSERT --                                                  7,51          Top
```


Τροποποίηση της ποσότητας μνήμης ενός από τα VM μας, με απευθείας επέμβαση στο αντίστοιχο αρχείο XML.

Προσέξτε ότι και στα δύο μπλοκ η μονάδα μέτρησης είναι το KiB. Πρόκειται για το Kibibyte (δεν έχει γίνει λάθος πληκτρολόγησης) και 1 KiB = 2^10 bytes = 1024 bytes. Στην πληροφορική βέβαια και το KB (Kilobyte) έχουμε συνηθίσει να το ταυτίζουμε με 1024 bytes, επειδή όμως γενικά ισχύει Kilo = 1000 είναι καλό να γράφουμε KiB κι όχι KB. Παρατηρούμε τώρα ότι η RAM του VM είναι 1048576KiB. Διαφορετικά, διαιρώντας με 1024, έχουμε 1024MiB. Το MiB είναι το λεγόμενο Mebibyte και ισχύει 1 MiB = 1024 KiB. Πολύ συχνά, στην καθομιλουμένη ή και στο γραπτό λόγο, λέμε ή γράφουμε Megabyte κι εννοούμε Mebibyte κι όλοι καταλαβαινόμαστε και χαιρόμαστε. Τώρα όμως που ανοίξαμε το CentOS.xml και πέσαμε πάνω στο KiB, πιστεύουμε ότι είναι καλό να θυμηθούμε τι *πραγματικά* σημαίνουν οι διάφορες μονάδες μέτρησης. Το VM μας, λοιπόν, έχει 1024MiB RAM ή αλλιώς 1GiB RAM (Gibibyte). Για να προσθέσουμε άλλο μισό GiB, τις δύο παρουσίες του 1048576 τις κάνουμε 1572864 (=1,5GiB). Αν θέλουμε να αφαιρέσουμε 256MB (για την ακρίβεια 256MiB), τότε τις δύο παρουσίες του 1048576 τις κάνουμε 786432 (=768MiB). Αποθηκεύουμε τις αλλαγές μας κι εγκαταλείπουμε τον editor. Την επόμενη φορά που θα ενεργοποιήσουμε το VM με το CentOS, π.χ., με ένα

```
cvar@thehost:~> sudo virsh start CentOS
```

η μνήμη του θα είναι αλλαγμένη ή, γενικότερα, θα ισχύουν οι όποιες τροποποιήσεις κάναμε προηγουμένως. Σημειώστε ότι αλλαγές στα αρχεία XML των μηχανών επιτρέπεται να κάνουμε ακόμη κι όταν αυτές είναι ενεργές. Απλά, μετά το "sudo virsh edit CentOS" (στη θέση του "CentOS" βάλτε το όνομα του VM σας) θα πρέπει να πληκτρολογούμε τα ακόλουθα:

```
cvar@thehost:~> sudo su
thehost:/home/cvar # cd /etc/libvirt/qemu
thehost:/etc/libvirt/qemu # virsh define CentOS.xml
Domain CentOS defined from CentOS.xml
thehost:/etc/libvirt/qemu # exit
exit
cvar@thehost:~>
```

Οι αλλαγές θα ληφθούν υπόψη την επόμενη φορά που το VM θα ενεργοποιηθεί (όχι μετά από reboot, αλλά μετά από shutdown και power-up). Να προσθέσουμε εδώ ότι όποτε θέλουμε να ελέγξουμε τα βασικά χαρακτηριστικά ενός VM, αρκεί να πληκτρολογούμε κάτι σαν

```
cvar@thehost:~> sudo virsh dominfo CentOS
Id:             -
Name:           CentOS
UUID:           41794bb2-c8e8-43aa-98c4-cfa64dfe2aab
OS Type:        hvm
State:          shut off
CPU(s):         2
Max memory:     524288 KiB
Used memory:    524288 KiB
Persistent:     yes
Autostart:      disable
Managed save:   no
Security model: apparmor
Security DOI:   0
cvar@thehost:~>
```

Άνω τελεία

Υπάρχουν πάρα πολλά ακόμα που θα μπορούσαμε να γράψουμε και να δείξουμε περί του virtualization με το KVM. Σκεφτείτε, π.χ., ότι δεν έχουμε πει τίποτε για το migration, το snapshotting ή το pool management. Το πιθανότερο είναι να επανέλθουμε. Προς το παρόν σας προτείνουμε ν' ασχοληθείτε κι εσείς. Για οτιδήποτε θελήσετε, μη διστάσετε να επικοινωνήσετε μαζί μας. Από τις όποιες συζητήσεις κάνουμε θα εξαρτηθεί σε μεγάλο βαθμό το πότε θα επανέλθουμε – βεβαίως, και τα θέματα που θα μας απασχολήσουν.


CMYK & WEB
cmykandweb.gr
GRAPHIC DESIGN
design • layout • web

Skill: Intermediate
Tags: Netflix, Kodi, Plex, media server, NFS, YaST, Handbrake, VLC

Φτιάξτε το δικό σας Netflix, με το Plex!

Οι πρόσφατες εξελίξεις με το Netflix μάς έβαλαν στην πρίζα. Χωρίς καμία καθυστέρηση ξεκινήσαμε να ψάχνουμε το όλο θέμα και γρήγορα συνειδητοποιήσαμε ότι, πράγματι, η παγκόσμια διάθεση του περιεχομένου της δημοφιλούς υπηρεσίας streaming, αν μη τι άλλο αποτελεί αξιοπρόσεκτη κίνηση. Σίγουρα δεν πρόκειται για κάποια ενδιαφέρουσα διαπίστωση από μέρους μας. Ό,τι ακολούθησε, πάντως, για εμάς τουλάχιστον *ήταν* ενδιαφέρον.

του Χρήστου Βαρελά

Εκεί που κάναμε τις διαπιστώσεις μας κι αναλογιζόμασταν αν και πόσο θα αλλάξει το ιντερνετικό τοπίο με ένα παγκόσμιο, πλέον, Netflix, επαληθεύσαμε μια υποψία μας: Το περιεχόμενο που θα παρέχεται σε διάφορες χώρες, αρχικά τουλάχιστον θα είναι ένα μικρό υποσύνολο του περιεχομένου που παρέχεται στην Αμερική. Κάπως έτσι βάλαμε στον πάγο την ιδέα για μηνιαία συνδρομή στην υπηρεσία. Στην επόμενη σκηνή είχαμε ήδη στήσει τον δικό μας media server και, απόλυτα ευχαριστημένοι με την ευχρηστία και τις επιδόσεις του, τον χαιρόμασταν στο τοπικό μας δίκτυο.

Δεν θα ήταν υπερβολή αν λέγαμε ότι με το κατάλληλο hardware, ένα καλό media server/player software και μια προσεκτικά συγκροτημένη συλλογή περιεχομένου, δικαιούται κάποιος να μιλά για το προσωπικό του Netflix. Είναι βέβαια αδύνατον να έχει στη διάθεσή του *όλο* το περιεχόμενο που σερβίρει η υπηρεσία, αλλά ας μην ξεχνάμε και τα ακόλουθα:

- κανείς στο Ηλιακό Σύστημα δεν νοιάζεται για *όλο* το περιεχόμενο που διαθέτει το Netflix
- λόγω πνευματικών δικαιωμάτων δεν διατίθεται το 100% του περιεχομένου σε κάθε χώρα
- η εταιρεία έχει αρχίσει να εντατικοποιεί τις πολιτικές μπλοκαρίσματος για όσους επιχειρούν πρόσβαση σε περιεχόμενο άλλης χώρας, π.χ., μέσω proxy ή VPN.

Επιστρέφοντας στα δικά μας, ας σημειώσουμε ότι οι σειρές που μας αρέσουν και παρακολουθούμε δεν είναι πολλές. Κάτι παρόμοιο συμβαίνει και με τις ταινίες, αφού απ' ό,τι φαίνεται τα τελευταία χρόνια έχουμε γίνει κάπως επιλεκτικοί. Κάθε φορά πάντως που θέλουμε να παρακολουθήσουμε ένα επεισόδιο μιας σειράς ή μία ταινία, μας αρέσει να έχουμε επιλογή για το πού θα την παρακολουθήσουμε (TV, tablet ή ακόμη και smartphone). Είναι επίσης επιθυμητός ένας κάποιος αυτοματοποι-

ημένος τρόπος οργάνωσης του περιεχομένου, π.χ., ανά τηλεοπτική σειρά και σεζόν. Τα δε metadata, όπως τα πόστερ ταινιών και σειρών, οι περιλήψεις, οι αξιολογήσεις κ.ο.κ., όταν υπάρχουν διευκολύνουν σημαντικά την οργάνωση αλλά και βελτιώνουν την όλη εμπειρία παρακολούθησης.

Αν το σκεφτούμε λίγο βλέπουμε όλες αυτές οι επιθυμίες ουσιαστικά είναι παράμετροι ενός γενικότερου προβλήματος – κι ευτυχώς λύσεις υπάρχουν. Και οι εν λόγω λύσεις υλοποιούνται με το κατάλληλο λογισμικό οργάνωσης, streaming ή/και αναπαραγωγής. Πράγματι, ένας δικτυωμένος υπολογιστής ή συσκευή που είναι εφοδιασμένος με κάποιον αξιοπρεπή media server, αφενός διατηρεί οργανωμένο το περιεχόμενό μας, αφετέρου είναι σε θέση να το στέλνει σε όποια άλλη συσκευή διαθέτει τον κατάλληλο player, ώστε να μιλά με τον server.

Δύο δημοφιλείς προτάσεις για home theater

Ο λόγος γίνεται για τα Kodi και Plex: δεν είναι παρόμοια σε λειτουργία, αλλά αμφότερα μπορούν να χρησιμοποιηθούν για τη δημιουργία ενός οικιακού συστήματος ψυχαγωγίας.

Το **Kodi** (http://kodi.tv), ο απόγονος του πάλαι ποτέ XBMC, είναι ένας media player Ανοικτού Λογισμικού που διανέμεται δωρεάν. Είναι εξαιρετικά ευέλικτο και προσαρμόσιμο και διατίθεται για Windows, Linux, OS X – ακόμη και για συσκευές με επεξεργαστή ARM, όπως είναι το Raspberry Pi. Το δε **Plex** (https://plex.tv) είναι ένας media server και player που αποτελεί ιδιόκτητο λογισμικό αλλά κι αυτός διανέμεται δωρεάν, εκτός από ορισμένες εκδοχές του player. Μέρος του κώδικα του Plex ξεκίνησε ως fork του XBMC, τον Μάιο του 2008. Το αποτέλεσμα αυτής της κοινής καταγωγής που μοιράζονται τα Kodi και Plex είναι ότι, πέρα από την αναπαραγωγή, αμφότερες οι εφαρμογές οργανώνουν αυτόματα το περιεχόμενό μας χρησιμοποιώντας μια ευέλικτη βάση δεδομένων και metadata, τα οποία συλλέγουν από online συστήματα όπως, π.χ., αυτά των thetvdb.com και themoviedb.org.

Τα Kodi και Plex έχουν όμως και σημαντικές διαφορές. Στο Kodi, για παράδειγμα, η βάση δεδομένων για την οργάνωση του περιεχομένου βρίσκεται σε κάθε client ή κεντρικά, σε έναν MySQL server. Από την άλλη, στο Plex η βάση συντηρείται πάντοτε κεντρικά, από τον Plex server. Το transcoding του ίδιου του περιεχομένου, ώστε να μπορεί ν' αναπαράγεται σε μια πληθώρα συσκευών με δυνατότητες και ισχύ που ποικίλουν, στην περίπτωση του Kodi γίνεται από τον client ενώ στην περίπτωση του Plex το αναλαμβάνει ο server. Για το Kodi εξάλλου διατίθεται μια υπερπληθώρα από add-ons, ενώ και οι δυνατότητες που έχει ο χρήστης για την τροποποίηση/προσαρμογή του περιβάλλοντος είναι πολύ μεγάλες. Απέναντι, στον κόσμο του Plex, η εικόνα είναι εντελώς διαφορετική: τα add-ons είναι λιγοστά, ενώ οι δυνατότητες για προσαρμογή του περιβάλλοντος είναι περιορισμένες.

Για το δικό μας setup στραφήκαμε στο Plex. Δεν μας ενδιαφέρει η πληθώρα των add-ons, τα οποία, σημειωτέον, μερικές φορές εγκαταλείπονται από τους προγραμματιστές τους και κατά συνέπεια δεν παίζουν σωστά σε επόμενες εκδόσεις του Kodi. Την ίδια στιγμή, μας αρέσει που η βάση δεδομένων του Plex media server είναι μία και τηρείται κεντρικά, ενώ και το transcoding γίνεται επίσης κεντρικά. Θεωρούμε εξάλ-

λου θετικό για το μέλλον του Plex το γεγονός ότι η εταιρεία πίσω του έχει συγκεκριμένο επιχειρηματικό πλάνο. Αναλυτικότερα, ο media server διανέμεται δωρεάν, το ίδιο και ο player για υπολογιστές με Windows, Linux, OS X, καθώς και για τα Apple TV και XBox. Από εκεί και πέρα, υπάρχει μια χρέωση της τάξης των 5 ευρώ για τους players συσκευών με iOS, Android, Windows Phone κ.ά. Επιπρόσθετα, για όποιον ενδιαφέρεται παρέχεται το προαιρετικό Plex Pass (https://plex.tv/subscription/about), με μηνιαία (4,99 ευρώ), ετήσια (39,99 ευρώ) ή Lifetime (149,99 ευρώ) χρέωση. Ο κάτοχος Plex Pass έχει έξτρα παροχές όπως, π.χ., offline playback για συγκεκριμένες συσκευές, λογαριασμούς χρηστών με περιορισμούς (βλ. parental controls), αυτόματο uploading φωτογραφιών από smartphones και tablets, τρέιλερς υψηλής ποιότητας και δωρεάν εφαρμογές αναπαραγωγής.

Στο υπόλοιπο του παρόντος άρθρου μάς απασχολεί το Plex.

Διαμορφώσεις για το σερβίρισμα περιεχομένου

Πού πρέπει να αποθηκεύονται τα media files μας; Στο ίδιο μηχάνημα με τον media server ή σε ξεχωριστό; Η απάντηση εξαρτάται από τον όγκο των αρχείων, όπως επίσης κι από το hardware που ήδη διαθέτουμε. Γενικά, θα μπορούσαμε να πούμε ότι υπάρχουν δύο βασικές προσεγγίσεις: εκείνη του Όλα σ' Ένα και Νοικοκυρεμένα (TM), καθώς κι εκείνη του Κατανεμημένα κι Ευλογημένα (TM). Εμείς επιλέξαμε μια τρίτη, ενδιάμεση προσέγγιση, πιο κοντά στην πρώτη. Πριν όμως πούμε περισσότερα, ας δούμε τις δύο βασικές προσεγγίσεις.

Όλα Σ' Ένα και Νοικοκυρεμένα (TM). Όλα τα αρχεία εικόνας/ήχου/βίντεο βρίσκονται στον ίδιο υπολογιστή ή NAS, όπου βρίσκεται κι ο Plex media server. (Σημειώστε ότι εκτός από τα βασικά OSes, ο Plex media server διατίθεται και για αρκετά μοντέλα NAS: https://plex.tv/downloads#pms-nas.) Η συγκεκριμένη προσέγγιση είναι αρκετά οικονομική, αφού δεν χρειαζόμαστε hardware *και* για τον file server *και* για τον media server. Έχει όμως και τα αρνητικά της: Αν το hardware δεν έχει την απαιτούμενη ισχύ, όπως συχνά συμβαίνει με τα NAS, είναι τότε πιθανό να παρατηρούνται προβλήματα κατά το playback – ειδικά όταν περισσότεροι από ένας πελάτες προσπαθούν να εξυπηρετηθούν.

Κατανεμημένα κι Ευλογημένα (TM). Τα αρχεία εικόνας/ήχου/βίντεο κατοικούν σε file server ή NAS και, μέσω NFS ή CIFS, μοιράζονται δικτυακά (και) προς το μηχάνημα στο οποίο τρέχει ο Plex media server. Το προαναφερθέν έχει την απαραίτητη ισχύ –ή τουλάχιστον πρέπει να την έχει– ώστε να ικανοποιεί έναν ή περισσότερους πελάτες ταυτόχρονα. Η συγκεκριμένη προσέγγιση είναι φανερά πιο ευέλικτη από την προηγούμενη. Είναι όμως και πιο ακριβή, ειδικά αν δεν διαθέτουμε file server ή NAS.

Όπως ήδη αναφέραμε, η δική μας προσέγγιση βρίσκεται ανάμεσα στις δύο βασικές – και πιο κοντά στην *Όλα Σ' Ένα και Νοικοκυρεμένα (TM)*. Το θέμα είναι ότι, με εξαίρεση ένα VM σε ρόλο Dropbox node (http://deltahacker.gr/?p=15441), αυτή την περίοδο δεν έχουμε κάποιον file server στο τοπικό δίκτυο. Έχουμε όμως έναν virtualization server με openSUSE (βλ. http://deltahacker.gr/?p=15388 και http://deltahacker.gr/?p=15425), ο οποίος είναι: **α)** ικανός να φιλοξενήσει τη σχετικά μικρή συλλογή από media files που διαθέτουμε στο host OS και **β)** αρκετά ισχυρός ώστε να φιλοξενήσει άλλο ένα

VM με Ubuntu Server και τον Plex media server. Το host OS (δηλαδή το openSUSE) μοιράζει τον κατάλογο με τα media files μέσω NFS, ενώ το Ubuntu Server VM με το Plex έχει προσαρτημένο το NFS exported directory με τα media files. Θα μπορούσαμε να εγκαταστήσουμε το Plex στο host OS κι όχι σε VM, όμως η ιδέα με το virtualization είναι ότι ο φυσικός υπολογιστής τρέχει τα απολύτως απαραίτητα κι από εκεί και πέρα έχουμε πολλά αυτόνομα κι εξειδικευμένα VMs. Υπό αυτή την έννοια, θα παρατηρούσε κάποιος ότι θα έπρεπε να έχουμε κι ένα VM σε ρόλο file server, με όλα τα media files. Έτσι είναι – και ίσως να το φροντίσουμε για το μέλλον. Προς το παρόν όμως η συλλογή με τα media files δεν είναι μεγάλη, ενώ η μνήμη RAM του φυσικού υπολογιστή δεν περισσεύει. Ακριβώς για τους λόγους αυτούς αποφασίσαμε να αποθηκεύσουμε τα media files *απευθείας* στον virtualization host – και να τα κάνουμε export μέσω NFS.

Ένα KVM VM με Ubuntu Server

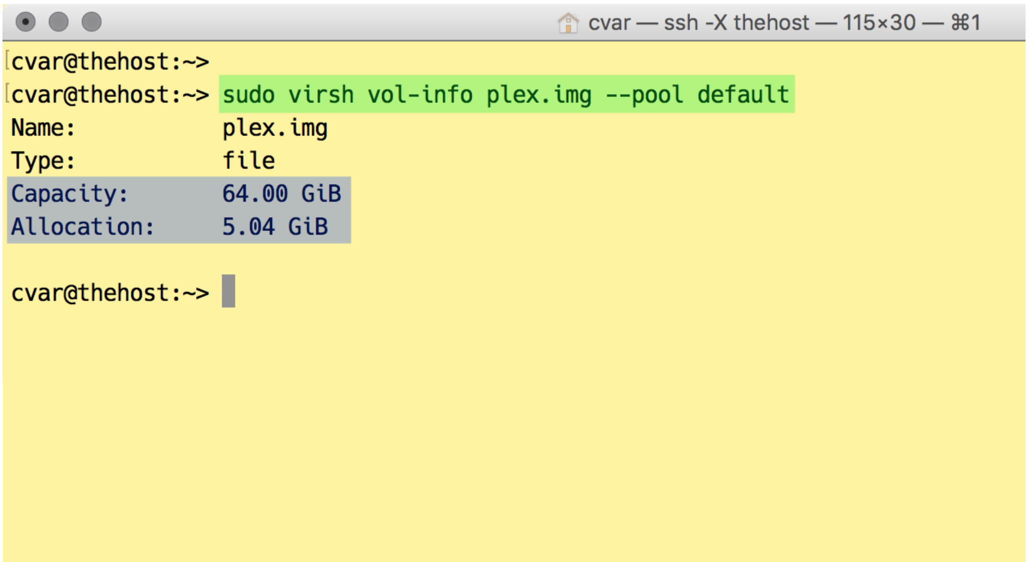
Όταν υπάρχει δυνατότητα επιλογής του λειτουργικού συστήματος για το μηχάνημα που θα φιλοξενήσει τον Plex media server, προτείνεται το Linux. Ο media server θα έχει καλύτερες επιδόσεις στο ίδιο –εικονικό ή φυσικό– hardware, όταν αντί για Windows ή OS X χρησιμοποιείται κάποια διανομή Linux. Φτιάξαμε λοιπόν ένα νέο KVM VM με 2GiB μνήμης RAM, δύο εικονικούς πυρήνες και sparse RAW file μεγέθους 64GiB για σκληρό δίσκο (βλ. και άρθρο στο http://deltahacker.gr/?p=15542).

```
cvar — ssh -X thehost — 115×30 — ⌘1
cvar@thehost:~>
cvar@thehost:~> sudo virsh dominfo Plex
Id:             5
Name:           Plex
UUID:           7fffd0ca-40c2-4f05-85a0-7783ad77a46a
OS Type:        hvm
State:          running
CPU(s):         2
CPU time:       19.8s
Max memory:     2097152 KiB
Used memory:    2097152 KiB
Persistent:     yes
Autostart:      enable
Managed save:   no
Security model: apparmor
Security DOI:   0

cvar@thehost:~>
```

Οι προδιαγραφές της εικονικής μηχανής που φιλοξενεί τον Plex media server. Παρατηρήστε, μεταξύ άλλων, ότι έχει 2 εικονικούς επεξεργαστές και 2GiB RAM. Έχει εξάλλου ενεργοποιηθεί το Autostart, που σημαίνει ότι το VM ξεκινά αυτόματα κατά το boot του host OS.

Παρεμπιπτόντως, το πραγματικό μέγεθος που καταλαμβάνει ο εικονικός δίσκος μετά την εγκατάσταση του Ubuntu Server, τις αναβαθμίσεις και την εγκατάσταση του Plex media server, είναι μόλις 5,04GiB.

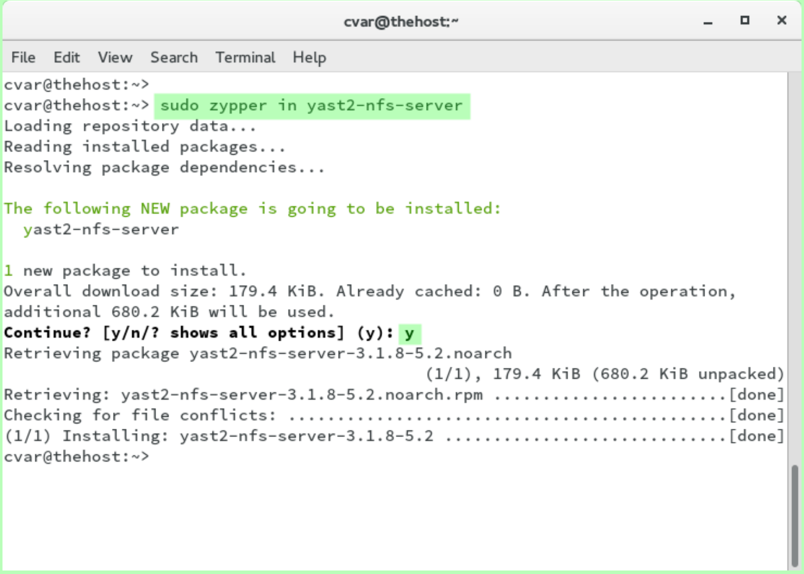


Το VM με το Ubuntu Server νομίζει ότι έχει έναν δίσκο μεγέθους 64GiB. Καλά κάνει και το νομίζει, ωστόσο αυτή τη στιγμή το αρχείο-δίσκος του VM καταλαμβάνει μόλις 5,04GiB στον αληθινό δίσκο του host computer.

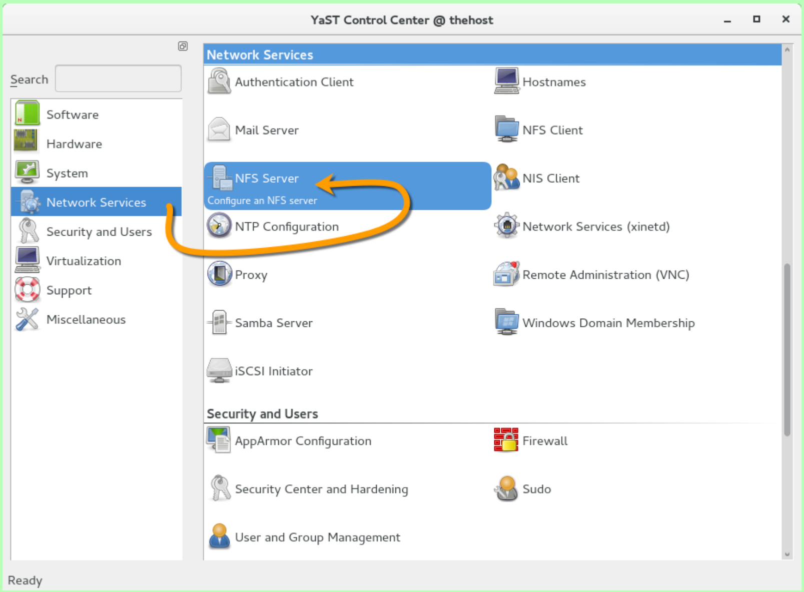
Η διαδικασία εγκατάστασης του Ubuntu Server –εμείς επιλέξαμε την έκδοση 14.04 LTS 64bit– είναι η τυπική και δεν αξίζει να σημειώσουμε κάτι. Απλά, αν θέλετε επιλέξτε από τον installer την ενεργοποίηση του OpenSSH server – αν κι αυτή η ενέργεια κάλλιστα μπορεί να γίνει και μετά την εγκατάσταση του λειτουργικού συστήματος.

Το openSUSE host ως NFS server

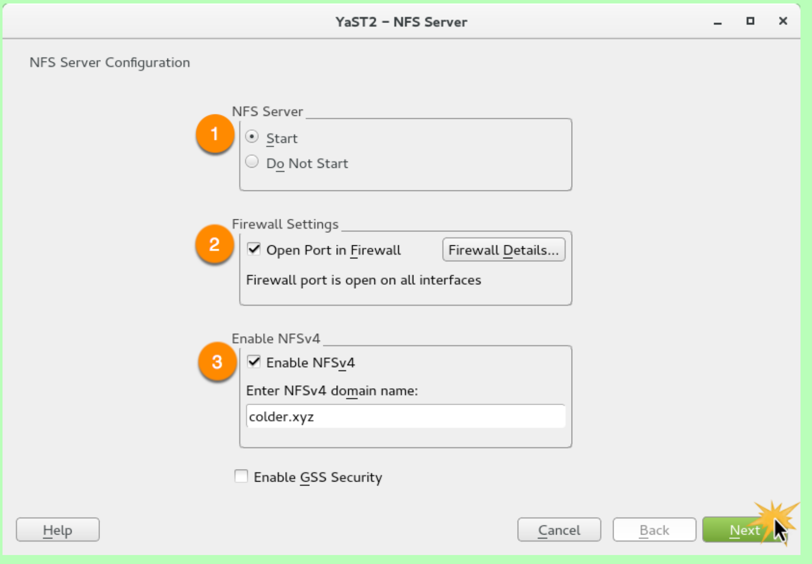
Βρισκόμαστε στο openSUSE, δηλαδή σε ένα σύστημα Linux, και θέλουμε να μοιράσουμε έναν κατάλογο με αρχεία (τα media files μας) στο Ubuntu Server, δηλαδή σε ένα άλλο σύστημα Linux. Με δεδομένα τα εμπλεκόμενα OSes, το καλύτερο πρωτόκολλο για τον διαμοιρασμό είναι το NFS (Network File System, https://en.wikipedia.org/wiki/Network_File_System). Αναλυτικότερα, θα κάνουμε export τον κατάλογο ~/Media από το openSUSE (εκεί κρατάμε τα media files) κι από το Ubuntu Server θα τον προσαρτήσουμε (mount) κάτω από τον κατάλογο /mnt/media. Στα screenshots που ακολουθούν –διαβάστε και τις αντίστοιχες περιγραφές– δείχνουμε πώς εργαστήκαμε για να εγκαταστήσουμε, να ρυθμίσουμε και να ενεργοποιήσουμε τον NFS server, καθώς και για να κάνουμε export αυτόν τον κατάλογο Media, από το home directory του χρήστη μας (cvar).



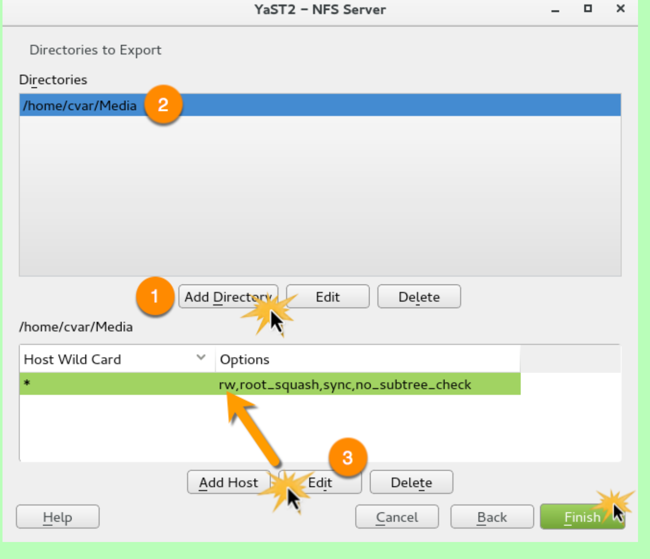
Για τη ρύθμιση του NSF server θα καταφύγουμε στο σχετικό άρθρωμα του YaST, το οποίο όμως απουσιάζει. Φροντίζουμε και το εγκαθιστούμε με το zypper, από τη γραμμή εντολών (το όνομα του πακέτου που χρειαζόμαστε είναι yast2-nfs-server).



Ξεκινάμε το YaST κι από τις κατηγορίες στη στήλη αριστερά, κάνουμε κλικ στο Network Services. Από τα αρθρώματα δεξιά, πατάμε πάνω στο NSF Server.



Θέλουμε ο NFS server να ξεκινά αυτόματα κατά την εκκίνηση του openSUSE (1) και τα απαραίτητα ports να μη μπλοκάρονται από το firewall (2). Ενεργοποιούμε επίσης και την υποστήριξη του NFSv4 (3). Η έκδοση 4 του NFS είναι βελτιωμένη τόσο ως προς την ασφάλεια, όσο κι ως προς τις επιδόσεις.



Στο νέο παράθυρο που εμφανίζεται κάνουμε ένα κλικ στο κουμπί Add Directory (1). Από το παράθυρο του file manager που ανοίγει, υποδεικνύουμε τον κατάλογο που πρόκειται να γίνει export. Για την περίπτωσή μας ο εν λόγω κατάλογος είναι ο /home/cvar/Media (2). Με το exported directory επιλεγμένο, στο πλαίσιο από κάτω βλέπουμε τις ιδιότητες που το χαρακτηρίζουν. Πατάμε στο κουμπί Edit και το μόνο που αλλάζουμε είναι το "ro", το οποίο κάνουμε "rw" (3). Έτσι, οι NFS clients που θα προσαρτούν το exported directory θα έχουν δικαίωμα τόσο για ανάγνωση, όσο και για εγγραφή. Ολοκληρώνουμε την εργασία μας με ένα κλικ στο κουμπί Finish, κάτω δεξιά.

Ένας τρόπος για να βεβαιωθούμε ότι ο κατάλογος /home/cvar/Media είναι πράγματι NFS exported, είναι ν' ανοίξουμε ένα τερματικό και να πληκτρολογήσουμε το ακόλουθο:

```
cvar@thehost:~> sudo showmount -e thehost
Export list for thehost:
/home/cvar/Media *
cvar@thehost:~>
```

(Το "thehost" είναι το άριστο hostname του virtualization host.)

Το Ubuntu VM ως NFS client

Συνδεόμαστε τώρα στο KVM VM με το Ubuntu Server, π.χ., μέσω SSH. Το username του χρήστη μας εκεί, είναι sub0. Φροντίζουμε ώστε το λειτουργικό σύστημα να είναι σε θέση να λειτουργεί ως NFS client. Τα πακέτα rpcbind και nfs-common πρέπει να είναι παρόντα, οπότε τα εγκαθιστούμε δίνοντας

```
sub0@plex:~$ sudo apt-get install rpcbind nfs-common
```

Ας δημιουργήσουμε τον κατάλογο υπό τον οποίο θα γίνει η προσάρτηση, μέσω NFS, του προαναφερθέντος exported directory:

```
sub0@plex:~$ mkdir -p ~/exported/Media
```

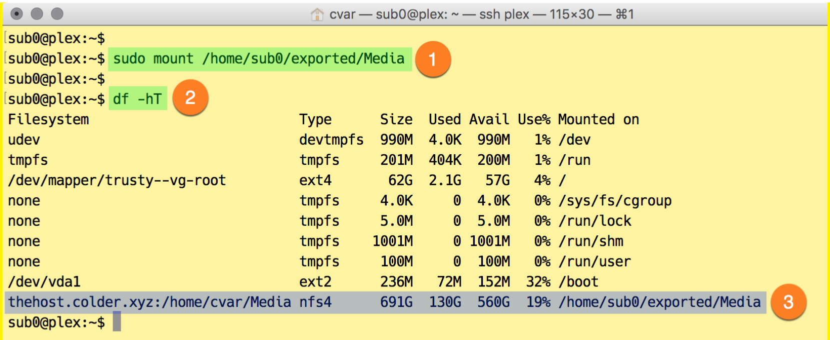
Με δικαιώματα υπερχρήστη ανοίγουμε το αρχείο /etc/fstab (π.χ., με ένα sudo nano /etc/fstab) και στο τέλος του προσθέτουμε την ακόλουθη γραμμή:

```
thehost.colder.xyz:/home/cvar/Media /home/sub0/exported/Media nfs rw,hard,intr 0 0
```

Στη θέση του "thehost.colder.xyz" θα βάλετε το FQDN ή το IP του NFS server σας, ενώ στις θέσεις των "/home/cvar/Media" και "/home/sub0/exported/Media" θα βάλετε τις διαδρομές του exported directory (στον NFS server) και του mount point (στον NFS client) αντίστοιχα. Μετά απ' αυτή την προσθήκη στο fstab, κάθε φορά που θα εκκινεί το λειτουργικό ο απομακρυσμένος κατάλογος θα προσαρτάται αυτόματα κατά NFS. Δεν χρειάζεται να κάνουμε reboot, αφού για το mounting αρκεί ένα

```
sub0@plex:~$ sudo mount /home/sub0/exported/Media
```

Προκειμένου να βεβαιωθούμε ότι όλα έγιναν σωστά μπορούμε, π.χ., να χρησιμοποιήσουμε το εργαλείο df.

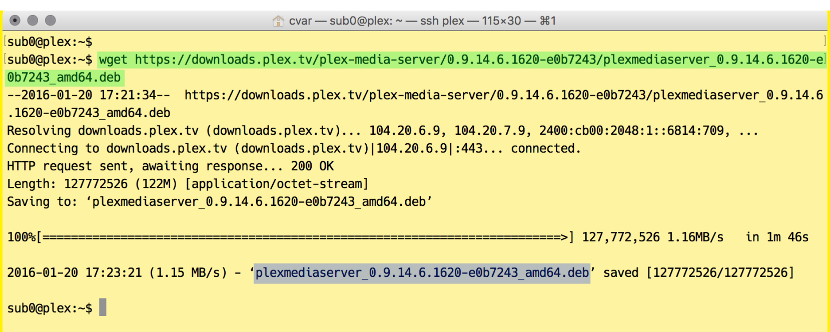


```
cvar — sub0@plex: ~ — ssh plex — 115×30 — ⌘1
sub0@plex:~$
sub0@plex:~$ sudo mount /home/sub0/exported/Media        1
sub0@plex:~$
sub0@plex:~$ df -hT        2
Filesystem                          Type      Size  Used Avail Use% Mounted on
udev                                devtmpfs  990M  4.0K  990M   1% /dev
tmpfs                               tmpfs     201M  404K  200M   1% /run
/dev/mapper/trusty--vg-root         ext4       62G  2.1G   57G   4% /
none                                tmpfs     4.0K     0  4.0K   0% /sys/fs/cgroup
none                                tmpfs     5.0M     0  5.0M   0% /run/lock
none                                tmpfs    1001M     0 1001M   0% /run/shm
none                                tmpfs     100M     0  100M   0% /run/user
/dev/vda1                           ext2      236M   72M  152M  32% /boot
thehost.colder.xyz:/home/cvar/Media nfs4      691G  130G  560G  19% /home/sub0/exported/Media        3
sub0@plex:~$
```

Προσαρτούμε το απομακρυσμένο exported directory μέσω NFS (1) και με τη βοήθεια του df (2) ελέγχουμε αν η διαδικασία πέτυχε. Η τελευταία γραμμή στην έξοδο του df φανερώνει πως όλα πήγαν καλά (3).

Εγκατάσταση Plex media server στο Ubuntu

Στο σημείο αυτό έχουμε το openSUSE σε ρόλο file server και όλα τα media files μοιράζονται, μέσω NFS, στο VM με το Ubuntu Server. Στο προαναφερθέν έχει έρθει η ώρα να εγκαταστήσουμε τον Plex media server, ο οποίος θα λειτουργεί ως άλλη μια υπηρεσία του συστήματος. Από τον web browser της προτίμησής μας πηγαίνουμε απευθείας στο https://plex.tv/downloads#pms-desktop. Στις τέσσερις καρτέλες πάνω πάνω, φροντίζουμε ώστε να είναι επιλεγμένη η "Linux". Παρατηρήστε ότι διατίθενται πακέτα για τρεις διαφορετικές διανομές Linux (Ubuntu, Fedora και CentOS). Επιπλέον, για κάθε διανομή διατίθενται δύο εκδοχές (64bit και 32bit). Εμείς τώρα θέλουμε το πακέτο για Ubuntu και συγκεκριμένα την εκδοχή 64bit. Κάνουμε δεξί κλικ πάνω στο αντίστοιχο κουμπί κι επιλέγουμε το "Copy Link" ή την αντίστοιχη λειτουργία του browser μας. Ουσιαστικά, αυτό που μας ενδιαφέρει είναι το πλήρες URL προς το πακέτο DEB του Plex media server, για Ubuntu Server 64bit. Στη συνέχεια συνδεόμαστε στο Plex μέσω SSH, αν δεν είμαστε ήδη συνδεδεμένοι, και κατεβάζουμε το πακέτο με τη βοήθεια του wget. Δείτε στα δύο ακόλουθα screenshots το downloading και την εγκατάσταση, με τη βοήθεια του dpkg.



```
cvar — sub0@plex: ~ — ssh plex — 115×30 — ⌘1
sub0@plex:~$
sub0@plex:~$ wget https://downloads.plex.tv/plex-media-server/0.9.14.6.1620-e0b7243/plexmediaserver_0.9.14.6.1620-e
0b7243_amd64.deb
--2016-01-20 17:21:34--  https://downloads.plex.tv/plex-media-server/0.9.14.6.1620-e0b7243/plexmediaserver_0.9.14.6
.1620-e0b7243_amd64.deb
Resolving downloads.plex.tv (downloads.plex.tv)... 104.20.6.9, 104.20.7.9, 2400:cb00:2048:1::6814:709, ...
Connecting to downloads.plex.tv (downloads.plex.tv)|104.20.6.9|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 127772526 (122M) [application/octet-stream]
Saving to: ‘plexmediaserver_0.9.14.6.1620-e0b7243_amd64.deb’

100%[==========================================================================>] 127,772,526 1.16MB/s   in 1m 46s

2016-01-20 17:23:21 (1.15 MB/s) - ‘plexmediaserver_0.9.14.6.1620-e0b7243_amd64.deb’ saved [127772526/127772526]

sub0@plex:~$
```

Αφού αντιγράψαμε το πλήρες URL προς την εκδοχή του πακέτου με τον Plex media server για τη διανομή μας, συνδεόμαστε στο Ubuntu Server μέσω SSH και κατεβάζουμε το σχετικό αρχείο με το wget.

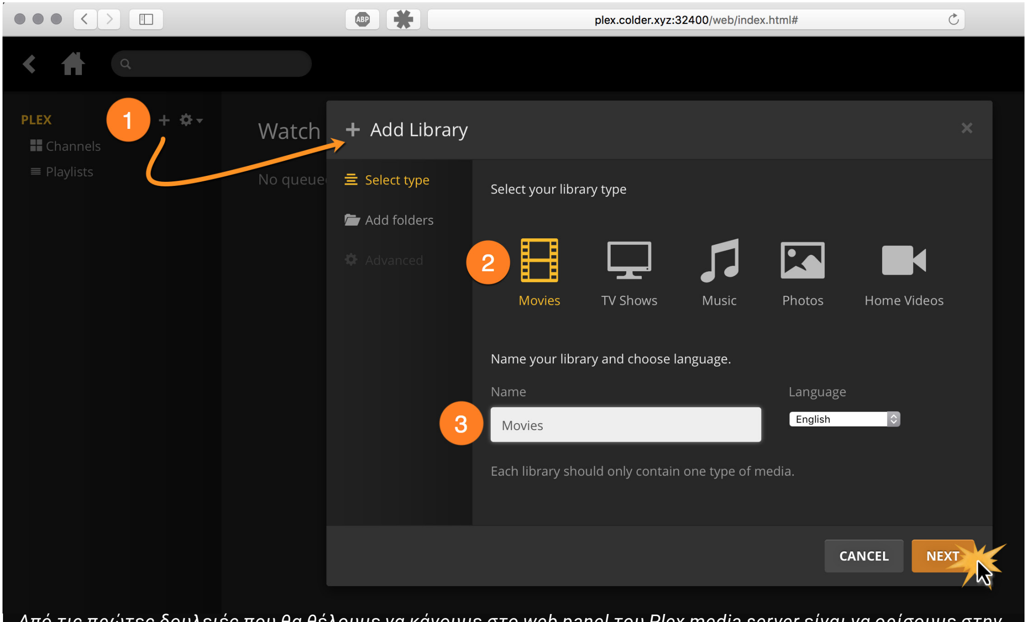
```
cvar — sub0@plex: ~ — ssh plex — 115×30 — ⌘1
sub0@plex:~$
sub0@plex:~$ sudo dpkg -i plexmediaserver_0.9.14.6.1620-e0b7243_amd64.deb
Selecting previously unselected package plexmediaserver.
(Reading database ... 86970 files and directories currently installed.)
Preparing to unpack plexmediaserver_0.9.14.6.1620-e0b7243_amd64.deb ...
Unpacking plexmediaserver (0.9.14.6.1620-e0b7243) ...
Setting up plexmediaserver (0.9.14.6.1620-e0b7243) ...
OK
plexmediaserver start/running, process 2892
Processing triggers for ureadahead (0.100.0-16) ...
Processing triggers for mime-support (3.54ubuntu1.1) ...
sub0@plex:~$
```

Για την εγκατάσταση του πακέτου DEB που μόλις κατεβάσαμε, επιστρατεύουμε το εργαλείο dpkg όπως φαίνεται στο screenshot. Η εγκατάσταση απαιτεί δικαιώματα διαχειριστή συστήματος, γι' αυτό και χρησιμοποιούμε το sudo.

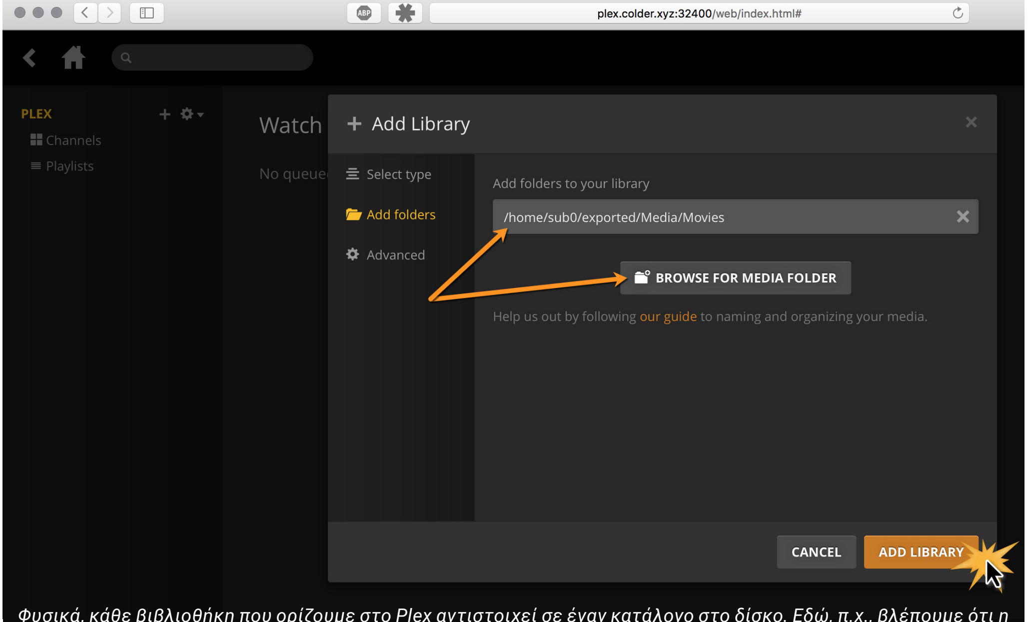
```
cvar — sub0@plex: ~ — ssh plex — 115×30 — ⌘1
sub0@plex:~$
sub0@plex:~$ sudo netstat -antp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:58276           0.0.0.0:*               LISTEN      2273/rpc.statd
tcp        0      0 0.0.0.0:1191            0.0.0.0:*               LISTEN      2963/Plex DLNA Serv
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      2073/rpcbind
tcp        0      0 0.0.0.0:32400           0.0.0.0:*               LISTEN      2899/Plex Media Ser
tcp        0      0 127.0.0.1:32401         0.0.0.0:*               LISTEN      2899/Plex Media Ser
tcp        0      0 0.0.0.0:32469           0.0.0.0:*               LISTEN      2963/Plex DLNA Serv
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      840/sshd
tcp        0      0 127.0.0.1:57880         0.0.0.0:*               LISTEN      2908/Plex Plug-in [
tcp        0      0 0.0.0.0:43546           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:56369         127.0.0.1:57880         CLOSE_WAIT  2963/Plex DLNA Serv
tcp        0      0 192.168.10.230:793      192.168.10.245:2049     ESTABLISHED -
tcp        0    188 192.168.10.230:22       192.168.10.5:60541      ESTABLISHED 2733/sshd: sub0 [pr
tcp6       0      0 :::42919                :::*                    LISTEN      -
tcp6       0      0 :::39304                :::*                    LISTEN      2273/rpc.statd
tcp6       0      0 :::111                  :::*                    LISTEN      2073/rpcbind
tcp6       0      0 :::22                   :::*                    LISTEN      840/sshd
sub0@plex:~$
```

Στην έξοδο του netstat παρατηρούμε ότι ο Plex media server έχει ήδη ενεργοποιηθεί. Τέλεια όλα!

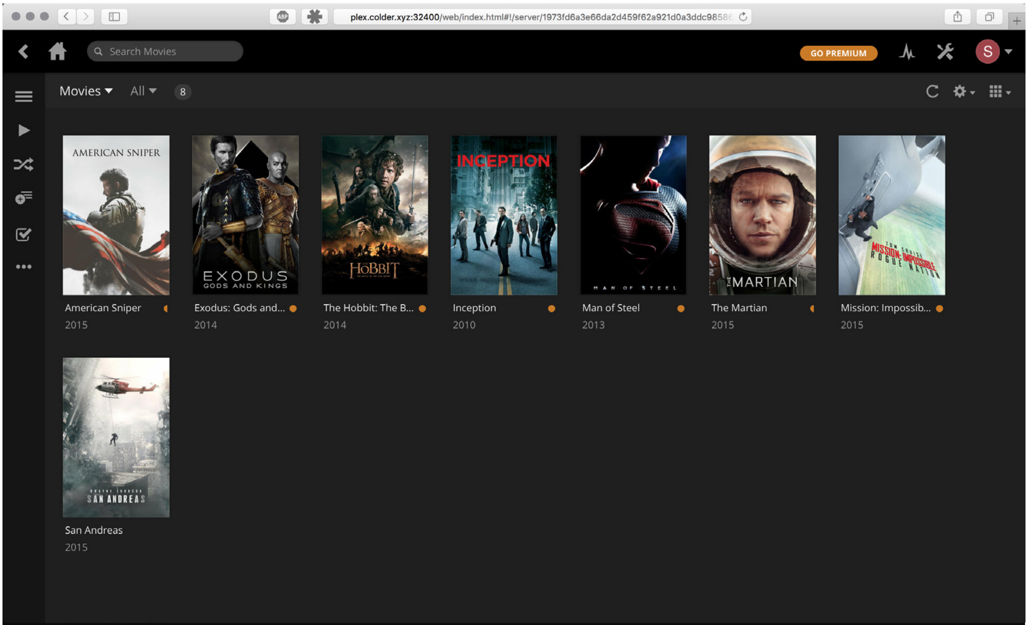
Η ρύθμιση και γενικά ο χειρισμός του Plex media server γίνονται από το web panel που παρέχει. Μέσα από το τοπικό μας δίκτυο, στο εν λόγω panel συνδεόμαστε από έναν οποιονδήποτε web browser και συγκεκριμένα επισκεπτόμενοι τη διεύθυνση http://plex.colder.xyz:32400/web. Φυσικά, στη θέση του "plex.colder.xyz" θα βάλετε είτε το FQDN του Plex box/VM είτε το IP του. Όπως θα διαπιστώσετε, το περιβάλλον που παρέχει το Plex είναι απλό, λειτουργικό κι αυτό που θα λέγαμε "διαισθητικό". Εκτός από κάποιες αρχικές ρυθμίσεις που δεν είναι απαραίτητες και μπορούν να γίνουν κι αργότερα, η πρώτη μας μέριμνα θα είναι να υποδείξουμε στο Plex φακέλους με περιεχόμενο που μπορεί να χειριστεί (βίντεο, φωτογραφίες, αρχεία ήχου). Δείτε μερικά χαρακτηριστικά screenshots και διαβάστε τις αντίστοιχες περιγραφές.



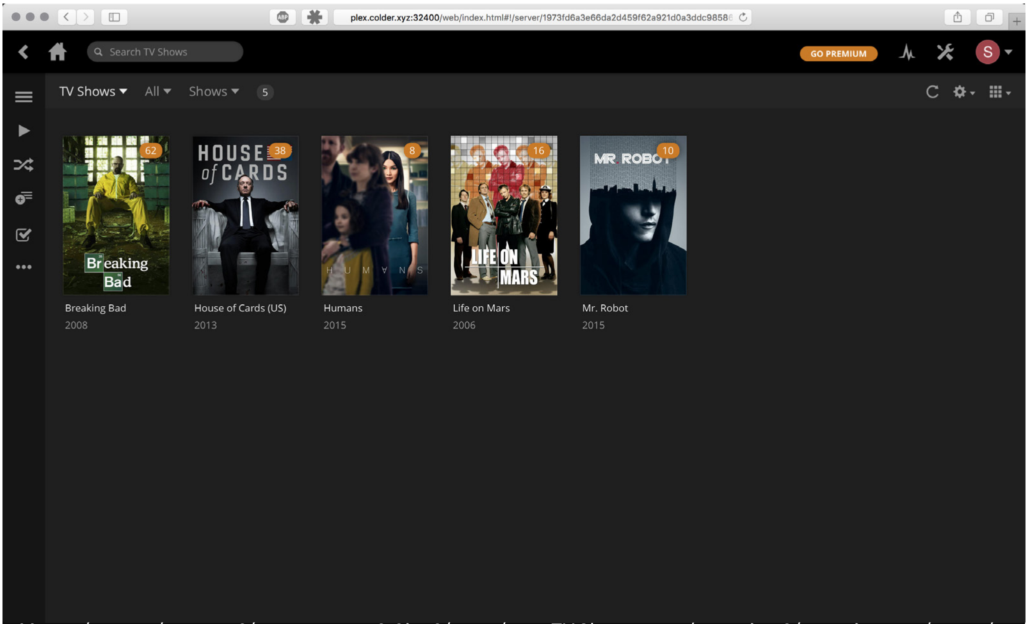
Από τις πρώτες δουλειές που θα θέλουμε να κάνουμε στο web panel του Plex media server είναι να ορίσουμε στην εφαρμογή βιβλιοθήκες με περιεχόμενο. Στο παράδειγμα βλέπουμε τον ορισμό βιβλιοθήκης με ταινίες (για τις δικές μας παραγωγές υπάρχει άλλη κατηγορία).



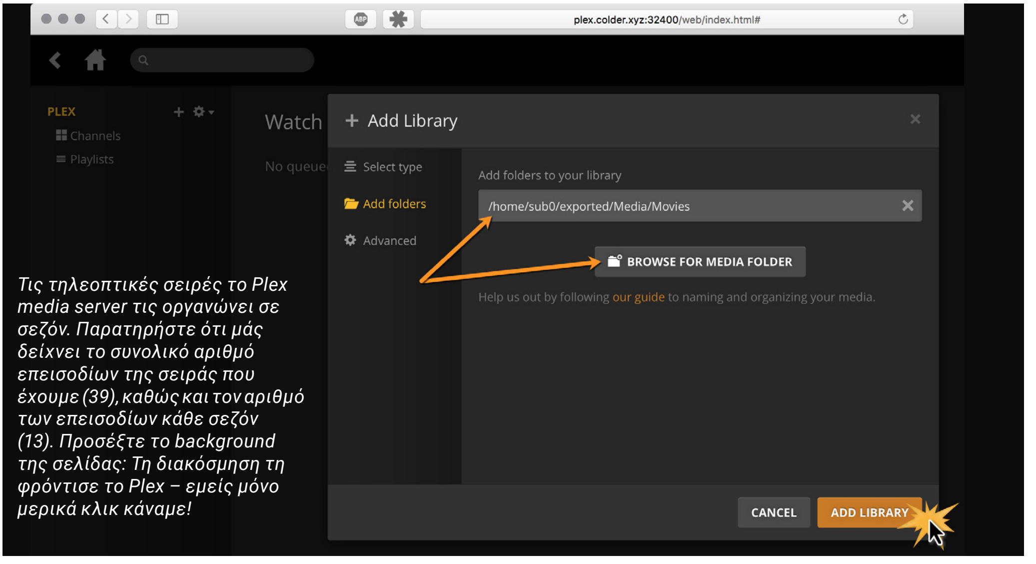
Φυσικά, κάθε βιβλιοθήκη που ορίζουμε στο Plex αντιστοιχεί σε έναν κατάλογο στο δίσκο. Εδώ, π.χ., βλέπουμε ότι η βιβλιοθήκη Movies αντιστοιχεί στον κατάλογο /home/sub0/exported/Media/Movies. Υπενθυμίζουμε ότι κάτω από τον /home/sub0/exported/Media είναι προσαρτημένο, μέσω NSF, ένα exported directory του openSUSE.



*Ιδού πώς φαίνονται τα περιεχόμενα της βιβλιοθήκης Movies, μετά από μερικές αναζητήσεις κι ενημερώσεις που *αυτομάτως* έκανε το Plex.*

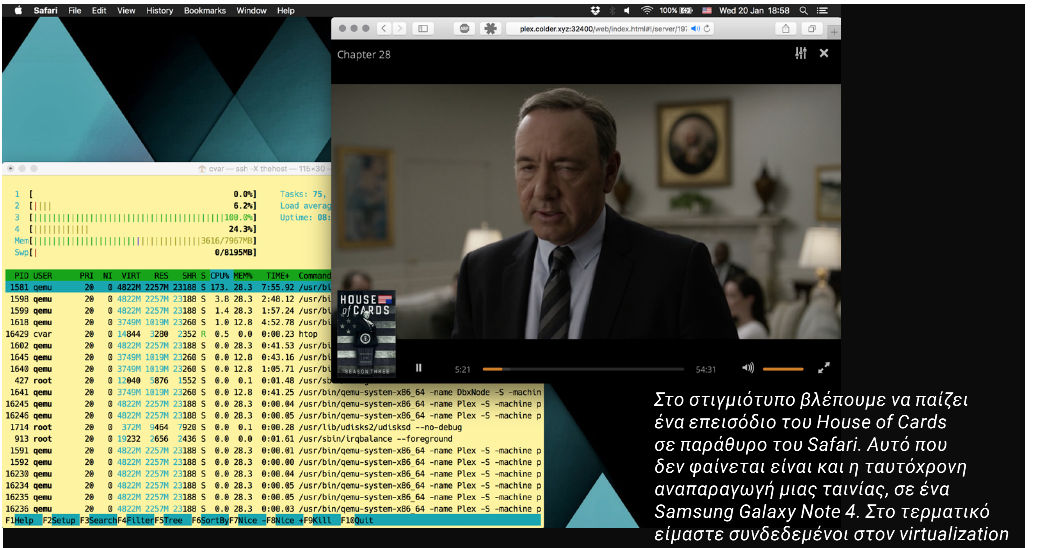


Με παρόμοιο τρόπο προσθέτουμε και μια βιβλιοθήκη ονόματι TV Shows, η οποία περιλαμβάνει τηλεοπτικές σειρές. Παρεμπιπτόντως, αυτή την περίοδο παρακολουθούμε την τρίτη σεζόν του House of Cards. Την είχαμε ξεκινήσει το καλοκαίρι, αλλά δεν κατορθώσαμε τότε να την ολοκληρώσουμε.



Τις τηλεοπτικές σειρές το Plex media server τις οργανώνει σε σεζόν. Παρατηρήστε ότι μάς δείχνει το συνολικό αριθμό επεισοδίων της σειράς που έχουμε (39), καθώς και τον αριθμό των επεισοδίων κάθε σεζόν (13). Προσέξτε το background της σελίδας: Τη διακόσμηση τη φρόντισε το Plex – εμείς μόνο μερικά κλικ κάναμε!

Η αναπαραγωγή του περιεχομένου που σερβίρει ο Plex media server γίνεται από έναν web browser, αφού πρώτα επισκεφθούμε τη διεύθυνση http://plex.colder.xyz:32400/web. Διατίθενται επίσης βολικοί players για Windows, OS X, iOS, Android και Windows Phone. Επιπρόσθετα, players υπάρχουν και για το Xbox, το Apple TV κι άλλες συσκευές. Ας υπογραμμιστεί εδώ ότι, όσον αφορά στο Apple TV, το Plex παίζει μόνο στις συσκευές 4ης γενεάς, δηλαδή στις πλέον πρόσφατες για τη στιγμή που γράφεται το παρόν.



*Στο στιγμιότυπο βλέπουμε να παίζει ένα επεισόδιο του House of Cards σε παράθυρο του Safari. Αυτό που δεν φαίνεται είναι και η ταυτόχρονη αναπαραγωγή μιας ταινίας, σε ένα Samsung Galaxy Note 4. Στο τερματικό είμαστε συνδεδεμένοι στον virtualization host και τρέχουμε το htop. Γενικά, το σύστημα συμπεριφερόταν ομαλότατα και είχαμε μόνο στιγμιαίες εξάρσεις στο CPU occupancy. Με δεδομένα **α)** τα ταπεινά χαρακτηριστικά του host computer (επεξεργαστής Core i3 με δύο πυρήνες, 8GiB μνήμης RAM), **β)** την ταυτόχρονη εκτέλεση δύο VMs και **γ)** το ταυτόχρονο streaming προς δύο συσκευές από το ένα εκ των δύο VMs, είμαστε σε θέση να πούμε ότι το Plex αποδίδει άριστα – ακόμη και σε σχετικά low-end hardware.*

VLC και Handbrake για το openSUSE

Από τη στιγμή που αρχίζουμε να αποθηκεύουμε media files στο openSUSE μας, είναι λογικό να θέλουμε και το VLC (http://www.videolan.org). Όχι απαραίτητα για να καθόμαστε μπροστά από τον virtualization host και να παρακολουθούμε σειρές και ταινίες, αλλά επειδή κάποιες φορές θα θέλουμε να ελέγξουμε στα γρήγορα ένα αρχείο video. Δεν ξέρουμε για εσάς, αλλά εμείς τουλάχιστον κάνουμε τέτοιους ελέγχους. Προκειμένου να εγκαταστήσουμε το VLC, οφείλουμε πρώτα να προσθέσουμε στο σύστημα το Packman. Πρόκειται για ένα αποθετήριο πακέτων (package repository) με χρήσιμα εργαλεία κι εφαρμογές, το οποίο δεν είναι παρόν μετά την εγκατάσταση του λειτουργικού αλλά μπορούμε να το προσθέσουμε "χειροκίνητα". Στο εν λόγω repository υπάρχει, μεταξύ άλλων, το υπέροχο κι εξαιρετικά χρήσιμο HandBrake (https://handbrake.fr), το οποίο λύνει τα χέρια εκείνων που έχουν ταινίες σε DVD και ψάχνουν έναν εύκολο τρόπο ώστε να τις μετατρέπουν σε αρχεία MP4, MKV ή οποιασδήποτε άλλης μορφής τους βολεύει.

Για την προσθήκη του Packman repository στο openSUSE, πληκτρολογούμε:

```
cvar@thehost:~> sudo zypper ar -f -n packman \
http://ftp.gwdg.de/pub/linux/misc/packman/suse/openSUSE_Leap_42.1/ packman
```

Μετά την προσθήκη του νέου αποθετηρίου, εγκαθιστούμε VLC και Handbrake ως ακολούθως:

```
cvar@thehost:~> sudo zypper in vlc vlc-codecs handbrake-cli handbrake-gtk
```

Το μόνο ζήτημα που ίσως αντιμετωπίσουμε με το HandBrake και τα δισκάκια DVD που διαθέτουμε, είναι ότι τα DVD drives στις μέρες μας σπανίζουν. Από την άλλη βέβαια μιλάμε για PC, οπότε, δεν μπορεί, μάλλον θα 'χει και DVD drive. Από το δικό μας PC το είχαμε αφαιρέσει εδώ και μήνες. Όταν όμως αποφασίσαμε να προσθέσουμε στη βιβλιοθήκη του Plex media server μια σειρά από αγαπημένες ταινίες σε DVD, χαρήκαμε που αντί να χαρίσουμε το DVD drive το είχαμε ξεχασμένο σ' ένα συρτάρι.